IBM Tivoli Workload Scheduler

Guia do Usuário e Referência

Versão 9 Release 2

S517-7781-15

IBM Tivoli Workload Scheduler

Guia do Usuário e Referência

Versão 9 Release 2

S517-7781-15

Nota
Antes de utilizar estas informações e o produto suportado por elas, leia as informações em "Avisos" na página 785.

Esta edição é aplicada à versão 9, liberação 2, nível de modificação 0 do Tivoli Workload Scheduler (número do programa 5698-WSH) e a todas as liberações e modificações subsequentes, até que seja indicado de outra forma em novas edições.

© Copyright IBM Corporation 1999, 2014.

Índice

Figuras ix

Tabelas xi

Sobre esta publicação xiii

O que Há de Novo nesta Liberação xiii
O Que Há de Novo nesta Publicação xiii
Quem Deve Ler Esta Publicação xiii
Publicações xiii
Acessibilidade xiv
Treinamento Técnico Tivoli xiv
Informações sobre suporte xiv
Convenções Utilizadas Nesta Publicação xiv
Convenções de Fontes xiv
variáveis e caminhos dependentes do sistema operacional xv
Sintaxe do Comando xv

Capítulo 1. Visão Geral do Tivoli Workload Scheduler 1

Entendendo Conceitos Básicos 1
Objetos de Banco de Dados do Tivoli Workload Scheduler 1
A Rede do Tivoli Workload Scheduler 19
Configurando seu Ambiente de Tempo de Execução do Tivoli Workload Scheduler 20
Definindo Atividades de Planejamento Usando o Tivoli Workload Scheduler 21
Controlando a Tarefa e o Processamento do Fluxo de Tarefas 22
Gerenciando Atividades de Planejamento de Produção com o Tivoli Workload Scheduler . . 25
Automatizando a Carga de Trabalho Usando Regras de Eventos 26
Interfaces com o Usuário do Tivoli Workload Scheduler 26
Iniciando a Produção 28

Capítulo 2. Entendendo Processos e Comandos Básicos 31

Emitindo Comandos em Sistemas Operacionais Windows 31
Processos da Estação de Trabalho do Tivoli Workload Scheduler 31
Iniciando e Parando Processos em uma Estação de Trabalho 37
Iniciando e parando oagent 38
Comunicação Interprocesso de Estação de Trabalho 38
Comunicação da Rede do Tivoli Workload Scheduler 40
Suporte para o Protocolo da Internet Versão 6 . . 42

Capítulo 3. Configurando o Ambiente de Tarefa 45

Visão Geral do Ambiente de Tarefa 45
Variáveis de Ambiente Exportadas por jobman . . 46
Customizando Formatação de Dados no stdlist 47
Customizando o Processamento da Tarefa em uma Estação de Trabalho UNIX - jobmanrc 48
Customizando a Seção MAIL_ON_ABEND de jobmanrc 49
Customizando o Processamento de Tarefas para um Usuário nas Estações de Trabalho UNIX - .jobmanrc . 49
Customizando o Processamento de Tarefa em uma Estação de Trabalho Windows - jobmanrc.cmd. . . 51
Customizando a Seção MAIL_ON_ABEND de jobmanrc.cmd 51
Customizando Processamento de Tarefa em uma Estação de Trabalho com o Windows - djobmanrc.cmd 52
Configurando Opções para Utilização das Interfaces com o Usuário 53

Capítulo 4. Gerenciando o Ciclo de Produção 57

Conceitos Básicos de Gerenciamento de Planos . . 57
Plano de Pré-Produção 59
Identificando Instâncias do Fluxo de Tarefas no Plano 60
Gerenciando Dependências de Acompanhamentos Externos para Tarefas e Fluxos de Tarefas 61
Plano de Produção 71
Entendendo as Opções de Transporte de Saldo 71
Plano de Experiência 73
Plano de Previsão 74
Customizando o Gerenciamento de Planos Utilizando Opções Globais 75
Criando e estendendo o plano de produção . . . 79
JnextPlan 81
Linha de Comandos Planman 82
Criando um Plano de Produção Intermediário . . 84
Criando um Plano Intermediário para uma Extensão de Plano 85
Recuperando informações sobre o plano de produção 86
Criando um Plano Trial 87
Criando um Plano Trial de uma Extensão de Plano de Produção 88
Criando um Plano de Previsão 89
Regras de Implementação 90
Desbloqueando o Plano de Produção 91
Reconfigurando o plano de produção. 91
Removendo o Plano de Pré-Produção. 92
Replicando os Dados do Plano no Banco de Dados 93

© Copyright IBM Corp. 1999, 2014

Monitorando a Replicação dos Dados do Plano no Banco de Dados 94
O comando stageman 95
Gerenciando Acessos Simultâneos ao Arquivo Symphony 96
Cenário 1: Acesso ao Arquivo Symphony Bloqueado por Outros Processos do Tivoli Workload Scheduler 96
Cenário 2: Acesso ao Arquivo Symphony Bloqueado pelo stageman. 97
Gerenciando Dependências Follows Utilizando Prompt de Transporte de Saldo 97
O comando logman 97
Iniciando o Processamento do Plano de Produção 100
Automatizando o processamento do plano de produção 101

Capítulo 5. Utilizando Garantia do Serviço de Carga de Trabalho 103
Ativando e Configurando a Workload Service Assurance 104
Planejando Tarefas Críticas 106
Processando e Monitorando Tarefas Críticas . . . 108
Cenário de Garantia de Serviço de Carga de Trabalho 110

Capítulo 6. Customizando a Carga de Trabalho Utilizando Tabelas de Variáveis 113
Migrando Parâmetros Globais de Versões Anteriores 114
A Tabela de Variáveis Padrão 114
Integridade de Dados para Tabelas de Variáveis 115
Mecanismo de Bloqueio para Tabelas de Variáveis 115
Segurança da Tabela de Variável 116
Resolução de Variável 116

Capítulo 7. Automação de Carga de Trabalho Orientada pelo Evento em Execução 119
O Processo de Gerenciamento de Regras de Eventos 122
Utilizando as Interfaces e Comandos Envolvidos 124
Definindo Regras de Eventos 126
Exemplos de Regras de Eventos 128
Notas sobre a operação de regras. 132
Elementos de Regra Acionados 133
Definindo Eventos Customizados. 134

Capítulo 8. Definindo Objetos no Banco de Dados 137
Definindo Objetos de Planejamento 137
Definição de estação de trabalho 139
Definição de classe de estação de trabalho. . . 153
Definição de domínio. 154
Definição de tarefa 156
Definição do Usuário. 192
Definição do calendário 197
Definição de Variável e Parâmetro 198
Definição da Tabela de Variáveis 202
Definição do Prompt 204
Definição do recurso 206
Definição do Grupo de Ciclos de Execução . . 207
Definição de fluxo de tarefas 216
Detalhes da Palavra-chave de Definição do Fluxo de Tarefas 220
Definição de Regra de Evento 260
Definindo os aplicativos de carga de trabalho . . 271
Criando um Modelo do aplicativo de carga de trabalho 272

Capítulo 9. Gerenciando Objetos no Banco de Dados - Composer 277
Configurando o Programa fe Linha de Comandos composer. 277
Configurando o Ambiente do Composer . . . 277
Executando o Programa composer 279
Executando Comandos a partir do Composer . . 281
Filtros e Curingas 281
Delimitadores e caracteres especiais 283
Códigos de Retorno do Composer 283
Composer commands 284
Verificação de Integridade Referencial 285
add. 287
authenticate 289
continue 290
delete 290
display 294
edit. 299
exit. 299
extract. 300
help 304
list 305
lock 310
modify 314
new 319
print 320
redo 321
rename 322
replace 324
system command 325
unlock. 326
validate 329
version 330

Capítulo 10. Gerenciando o aplicativos de carga de trabalho . . . 331
Resolvendo o Arquivo de Mapeamento. 331
Implementando um aplicativo de carga de trabalho 334
Comando wappman 335

Capítulo 11. Gerenciando Objetos no Plano - conman 339
Configurando o Programa de linha de Comandos conman 339
Configurando o Ambiente do conman 339
Executando conman 341
Executando comandos a partir do conman . . . 343
Curingas 343

Delimitadores e caracteres especiais 343
Processamento de comandos do conman . . . 344
Selecionando Tarefas em Comandos 345
Sintaxe 345
Argumentos 345
Selecionando Fluxos de Tarefas em Comandos . . 353
Sintaxe 353
Argumentos 354
Gerenciando Tarefas e Fluxos de Tarefas a partir de Agentes de Nível Anterior 360
Conman commands 360
adddep job 362
adddep sched 364
altpass. 366
altpri 367
bulk_discovery 368
cancel job. 368
cancel sched 370
checkhealthstatus 371
confirm 372
console 373
continue 374
deldep job 374
deldep sched 376
deployconf 377
display 378
exit 380
fence 380
help 382
kill 383
limit cpu 384
limit sched 385
link. 386
listsym 388
recall 390
redo 391
release job 392
release sched 393
reply 395
rerun 396
resetFTA 398
resource 399
setsym 400
showcpus 401
showdomain 408
showfiles 409
showjobs 411
showprompts 427
showresources 429
showschedules 431
shutdown 436
start 437
startappserver 439
startbrokerapp 440
starteventprocessor 440
startmon 441
status 442
stop 442
stop ;progressive 444
stopappserver 445
stopbrokerapp 447
stopeventprocessor 447
stopmon 448
submit docommand 449
submit file 452
submit job 456
submit sched 459
switcheventprocessor 463
switchmgr 464
system command 465
tellop 466
unlink 466
version 469

Capítulo 12. Ativando recursos de planejamento dinâmico em seu ambiente 471
Vantagens do tipos de tarefas com opções avançadas 473
Criando o tipos de tarefas com opções avançadas 474
Códigos de Retorno 476
Definindo Variáveis e Senhas para Resolução Local em Agentes Dinâmicos 477
Especificando Variáveis Locais e Senhas nas Definições de Tarefa 478
Usando Variáveis em Tarefas do Dynamic Workload Broker 480
Passando Variáveis Entre Tarefas na Mesma Instância do Fluxo de Tarefas 480
Definindo Relacionamentos de Afinidade 486
Promovendo Tarefas Programadas em Conjuntos Dinâmicos 486
Incluir Capacidades Dinâmicas em Tarefas do Tivoli Workload Scheduler Existentes 487
Um Cenário de Negócios 487
Cenário: Criando uma Definição de Tarefa e Enviando para um Conjunto Dinâmico 489
Cenário: Criando uma Definição de Tarefa e Enviando para um Conjunto 490
Limitações para tarefas no fluxo de tarefas **USERJOBS** no planejamento dinâmico 490

Capítulo 13. Utilizando Comandos Utilitários 493
Descrições dos comandos 493
at e batch. 494
cpuinfo 497
datecalc 499
datamigrate 503
Excluir 505
evtdef 506
evtsize. 509
jobinfo. 511
jobstdl. 513
maestro 515
makecal 516
metronome 518
morestdl 518
parms 520
release. 522
rmstdlist 523

sendevent 524
showexec 526
shutdown 527
ShutDownLwa 527
StartUp 527
StartUpLwa 528
tws_inst_pull_info 529
version 529
Comandos não suportados 530

Capítulo 14. Usando os comandos do utilitário no ambiente dinâmico. . . . 533
Arquivo de Configuração da Linha de Comandos 533
exportserverdata 537
importserverdata 539
jobprop 540
movehistorydata 541
param 543
recurso 546
Utilizando o comando resource a partir de um agente 554
sendevent 556
twstrace 557

Capítulo 15. Obtendo Relatórios e Estatísticas 559
Configuração para Utilizar Comandos de Relatório 559
Alterando o formato de data 560
Descrições dos comandos 560
rep1 - rep4b 561
rep7 562
rep8 563
rep11 564
reptr 565
xref 567
Amostra de saídas do relatório 568
Relatório 01 - Listagem de Detalhes da Tarefa: 568
Relatório 02 - Listagem de Avisos: 570
Relatório 03 - Listagem de Calendários: . . . 570
Relatório 04A - Listagem de Parâmetros: . . . 571
Relatório 04B - Listagem de Recursos: 571
Relatório 07 - Listagem de Históricos da Tarefa: 572
Relatório 08 - Histograma da Tarefa:. 572
Relatório 9B - Detalhes da Produção Planejada: 573
Relatório 10B - Detalhes da Produção Real: . . 574
Relatório 11 Agendamento de Produção Planejada: 574
Relatório 12 - Relatório de Referências Cruzadas: 576
Programas de Extração de Relatório 578
jbxtract 578
prxtract 579
caxtract 580
paxtract 580
rextract 581
r11xtr 582
xrxtrct 583
Executando Relatórios do Dynamic Workload Console e relatórios em lote 587
Relatórios de Histórico 589
Relatórios de Produção 590
Executando o relatórios em lote a partir da Interface da Linha de Comandos 591

Capítulo 16. Gerenciando fusos horários. 597
Ativando o Gerenciamento de Fuso Horário . . . 597
Como o Tivoli Workload Scheduler Gerencia Fusos Horários 598
Mudando para o Horário de Verão Ativado . . . 600
Mudando para o Horário de Verão Desativado . . 600
Regras Gerais 600

Capítulo 17. Definindo Métodos de Acesso para Agentes 603
Interface do método de acesso. 604
Sintaxe da linha de comandos do método . . . 604
Mensagens de resposta do método 606
Arquivo de opções do método. 607
Executando Métodos 609
Tarefa LJ (Launch Job) 610
Tarefa MJ (Manage Job) 610
Apenas agentes estendidos da tarefa CF (Check File) 611
Apenas agentes estendidos da tarefa GS (Get Status). 611
Comando Cpuinfo Apenas para Agentes Estendidos 612
Resolução de Problemas. 612
Mensagens de erro da lista padrão de tarefas 612
Método não executável 612
Mensagens do Console Manager Apenas para Agentes Estendidos 613
Mensagens do Composer e do Compilador Apenas para Agentes Estendidos 613
Mensagens Jobman Apenas para Agentes Estendidos 613

Capítulo 18. Gerenciando Dependências da Interligação de Redes 615
Visão Geral das Dependências da Interligação de Redes 615
Compreendendo Como uma Dependência entre Redes É Mostrada 616
Configurando um agente de rede. 617
Uma Amostra da Definição do Agente de Rede 618
Definindo uma Dependência de Interligação de Redes 619
Gerenciando Dependências da Interligação de Redes no Plano. 620
Estados de Tarefas Definidas no Fluxo de Tarefas EXTERNAL 620
Trabalhando com Tarefas Definidas no Fluxo de Tarefas EXTERNAL 621
Amostra de Cenários de Gerenciamento da Dependência entre Redes 621
Dependências da interligação de redes em um ambiente misto 623

Capítulo 19. Definindo e Gerenciando Dependências Cruzadas 625
Uma Introdução a Dependências Cruzadas . . . 625
Fluxo de Processamento no Ambiente de Planejamento Distribuído 627
Definindo uma Dependência Cruzada 629
Monitorando uma Resolução de Dependência Cruzada no Plano de Produção 631
Como o Status da Tarefa Sombra Muda até uma Conexão Ser Estabelecida 631
Como o Status da Tarefa Sombra Muda após a Conexão Ser Estabelecida 637
Como Ver Por que o Status da Tarefa Sombra É FAIL 638
Status da Tarefa Sombra durante a Recuperação ou Nova Execução da Tarefa Remota 638
Como Transporte se Aplica a Dependências Cruzadas 639
Gerenciando Tarefas Sombra no Plano de Produção 639

Capítulo 20. Gerenciando um Ambiente Dinâmico do IBM i 641
Definindo Agentes em Sistemas IBM i 641
Definindo Tarefas em Sistemas IBM i 641
Gerenciando Agentes em Sistemas IBM i 642
Iniciando e Parando Agentes em Sistemas IBM i 642
Usando Comandos Utilitários para Agentes em Sistemas IBM i 642
Planejando Tarefas em Sistemas IBM i 643
O log da tarefa do agente e a variável de ambiente TWSASPOOLS 643
Monitoramento de Tarefas-filhas em Agentes IBM i 644
A recuperação do código de retorno do agente 646
Controlando o Ambiente de Tarefa com o Código de Retorno do Usuário 647
Método Alternativo para Configurar o Código de Retorno do Usuário 648

Apêndice A. Evento de Automação e Definições de Ação da Carga de Trabalho Orientada a Evento 651
Provedores e Definições de Eventos 651
Eventos do TWSObjectsMonitor 651
Eventos de FileMonitor 654
Eventos do TWSApplicationMonitor. 659
Provedores e Definições de Ação 661
Ações de GenericAction 661
Ações de MailSender 662
Ações de MessageLogger 662
Ações SmartCloud Control Desk 662
Ações TBSMEventForwarder 663
Ações do TECEventForwarder. 663
Ações de TWSAction 664
TWSForZosAction 664

Apêndice B. Referência do Esquema Job Submission Description Language 667
Elementos JSDL 672
Recursos na Definição de Tarefa 707

Apêndice C. Referência rápida para comandos 709
Gerenciando o Plano 709
Gerenciando objetos no banco de dados 710
Comandos de Objetivo Geral 710
Objetos de Planejamento 710
Comandos do Composer 715
Gerenciando objetos no plano 718
Comandos do Conman 718
Comandos utilitários 722
Comandos de relatório 724

Apêndice D. Definindo e Gerenciando Tarefas de Ramificação Genéricas . . 727
Introdução 727
Terminologia 728
Recursos da Tarefa de Ramificação 731
Vantagens da Tarefa de Ramificação 732
Cenários de Amostra 733
Cenários Baseados em Tipo de Condição . . . 733
Cenários Baseados no Tipo de Ação 757
Cenário de Ação de Sinal 763
Trabalhando com a Tarefa de Ramificação 766
Pré-requisitos para Executar Tarefas de Ramificação no Windows 766
Definindo a Tarefa de Ramificação e a Tarefa de Sinal no Banco de Dados 766
Colocando a Tarefa de Ramificação no Fluxo de Tarefas 768
Usando a Tarefa ABEND 769
Especificando os Parâmetros da Tarefa de Ramificação 770
Referência de Parâmetros 770
Distinção entre Maiúsculas e Minúsculas . . . 774
Exemplos de Condições de Amostra 775
Notas Importantes sobre a Tarefa de Ramificação 780

Apêndice E. Acessibilidade 783

Avisos 785
Marcas Registradas 786

Índice Remissivo 789

Figuras

1. Rede de Domínio Único 16
2. Rede de domínios múltiplos 17
3. Árvore de Processos no UNIX 35
4. Árvore de Processos no Windows 36
5. Comunicação entre os Processos 40
6. Critérios de Correspondência do Mesmo Dia 61
7. Critérios de Correspondência Anteriores Mais Próximos 62
8. Nos Critérios de Correspondência de um Intervalo Relativo 62
9. Nos Critérios de Correspondência de um Intervalo Absoluto 63
10. Tarefa Predecessora Anterior Mais Próxima 64
11. Instância do Predecessor Pendente 65
12. Critérios de Correspondência no Mesmo Dia - Etapa 1: no Início do Dia (SOD) em uma Quinta-feira 66
13. Critérios de Correspondência no Mesmo Dia - Etapa 2: às 9h 66
14. Critérios de Correspondência no Mesmo Dia - Etapa 3: às 15h 67
15. Critérios de Correspondência Anteriores Mais Próximos - Etapa 1: antes da 8h 67
16. Critérios de Correspondência Anteriores Mais Próximos - Etapa 2: às 8h em Dias da Semana, Exceto Quintas-feiras e Sextas-feiras 68
17. Critérios de Correspondência Anteriores Mais Próximos - Etapa 3: às 9h nas Quintas-feiras e Sextas-feiras 68
18. Critérios de Correspondência Anteriores Mais Próximos - Etapa 4: às 15h Todo Dias 68
19. Critérios de Correspondência de Intervalo Relativo - no Início do Dia na Quinta-feira . . 70
20. Critérios de Correspondência de Intervalo Absoluto - no Início do Dia na Quinta-feira . . 71
21. Caminho crítico 109
22. Definição do Usuário 196
23. Links da Rede 387
24. Rede de Exemplo 439
25. Rede de Exemplo 444
26. Rede de Exemplo 445
27. Estações de Trabalho de Rede Desvinculadas 468
28. Exemplo quando a conversão de início do dia não é aplicada 599
29. Exemplo quando a conversão de início do dia é aplicada. 599
30. Redes local e remota 618
31. Lógica da Dependência Cruzada 627
32. Transição de Status da Tarefa Sombra até a Conexão Ser Estabelecida 631
33. Instância a ser Conectada se o Horário Planejado da Tarefa Sombra Estiver Incluído no Intervalo do CP. 634
34. Instância a Ser Conectada se a Instância que Precede de Forma Mais Próxima o Horário Planejado da Tarefa Sombra Existe no LTP, mas Foi Cancelado no CP 635
35. O Horário Planejado da Tarefa Sombra Está Incluído no CP, mas Não Existe Nenhuma Instância para Conectar 636
36. A Instância a Ser Conectada Existe, mas Ainda Não Está Incluída no CP 636
37. O Intervalo do LTP Ainda Não Contém o Horário Planejado da Tarefa Sombra 637
38. Cadeia de Transição de Status da Tarefa Sombra após a Conexão Ter Sido Estabelecida 638
39. Propósito da tarefa de ramificação 728
40. Termos relacionados à definição de fluxo de tarefas 730
41. Termos relacionados à execução do fluxo de tarefas (instância do fluxo de tarefas concreta) 731
42. Definição de ramificação simples (SUCC) 734
43. Status final de ramificação simples (SUCC) 734
44. Definição de ramificação simples (ABEND) 735
45. Definição de ramificação longa (SUCC) 737
46. Status final (ABEND) da ramificação longa 738
47. Diversas Tarefas de Ramificação em um Fluxo de Tarefas. 740
48. Definição de encerramento anormal do pai (SUCC) 741
49. Cenário de Padrão - Definição 744
50. Definição de cenário de padrão negado 746
51. Padrão na Definição de Linha de Padrão 748
52. Definição de padrão na linha de padrão negada. 750
53. Definição de ramificação de comparação numérica 753
54. Definição de condição complexa 755
55. Definição de Ações Pausar e Liberar 759
56. Definição de cenário de diversas ações pausar e liberar 762
57. Definição de ação de sinal 764

© Copyright IBM Corp. 1999, 2014

Tabelas

1. Sintaxe do Comando xv
2. Cenário 1. Não há restrição de tempo no grupo do ciclo de execução 8
3. Cenário 2. Restrição de tempo no grupo do ciclo de execução sem deslocamento 9
4. Cenário 3. Restrição de tempo no grupo do ciclo de execução com deslocamento (+1 12:00) . 9
5. Dica 18
6. Iniciando e Parando oTivoli Workload Scheduler em uma Estação de Trabalho . . . 37
7. Iniciando e parando oagent 38
8. Variáveis de Ambiente de Job para o Windows 46
9. Variáveis de Ambiente de Job para o UNIX 47
10. Variáveis Definidas por Padrão no Arquivo jobmanrc 48
11. Variáveis Definidas por Padrão no Arquivo jobmanrc.cmd 51
12. Configurações das Opções Globais do Saldo Transportado 72
13. Configurações Resultantes do Transporte de Saldo. 72
14. Opções Globais de Garantia de Serviço de Carga de Trabalho 104
15. Opções Locais de Garantia de Serviço de Carga de Trabalho 105
16. O Relacionamento entre as Tabelas de Variáveis em suas Variáveis no Arquivo de Segurança do Tivoli Workload Scheduler . . 116
17. Comandos conman para Gerenciar Mecanismos de Monitoramento 123
18. Comandos conman para Gerenciar o Servidor de Processamento de Eventos 124
19. Interfaces e Comandos para Gerenciar a Automação de Carga de Trabalho Orientada a Evento 125
20. Lista de Palavras-chave de Objetos de Planejamento Suportadas 138
21. Lista de Palavras Reservadas ao Definir Tarefas e Fluxos de Job 138
22. Lista de Palavras Reservadas ao Definir Estações de Trabalho 139
23. Lista de Palavras Reservadas ao Definir Usuários 139
24. Configurações de Atributos para Tipos de Estação de Trabalho de Gerenciamento . . . 140
25. Configurações de Atributos para Tipos de Estação de Trabalho de Destino 141
26. Tipo de Comunicação, Dependendo do Valor do Nível de Segurança 150
27. Exemplos: Renomeando a Definição de Tarefa 157
28. Operadores de Comparação. 160
29. Operadores lógicos. 160
30. Opções e Ações de Recuperação 161
31. Atributos necessários e opcionais para a definição de uma tarefa de serviços da web . 166
32. Atributos Necessários e Opcionais para a Definição de uma Tarefa de Transferência de Arquivos 168
33. Atributos necessários e opcionais para a definição de uma tarefa de Provisioning . . 173
34. Atributos Obrigatórios e Opcionais para a Definição de uma Tarefa J2EE. 176
35. Atributos Necessários e Opcionais para a Definição de uma Tarefa de Banco de Dados . 177
36. Atributos Obrigatórios e Opcionais para a Definição de uma Tarefa MSSQL. 180
37. Atributos Obrigatórios e Opcionais para a Definição de uma Tarefa Java. 181
38. Atributos Obrigatórios e Opcionais para a Definição de uma Tarefa Executável. 182
39. Atributos necessários e opcionais para a definição de uma tarefa de comando remoto . 183
40. Atributos Obrigatórios e Opcionais para a Definição de uma Tarefa de Método de Acesso 185
41. Os atributos necessários e opcionais para a definição de uma tarefa z/OS. 187
42. Atributos Obrigatórios e Opcionais para a Definição de uma Tarefa do IBM i. 187
43. Atributos necessários e opcionais para a definição de uma tarefa do OSLC Automation. 188
44. Atributos necessários e opcionais para a definição de uma tarefa do OSLC Provisioning.. 190
45. Como manipular uma barra invertida em substituição de variável. 199
46. Palavras-chave que podem utilizar parâmetros locais em comandos **submit** . . . 200
47. Palavra-chave de Acesso Necessárias em Tabela de Variável o Arquivo de Segurança (Objeto vartable) e Ações Permitidas. . . . 204
48. Lista de Palavras-chave de Planejamento 217
49. Explicação da Notação que Define o Número de Ocorrências para um Elemento de Linguagem. 260
50. Eventos `TWSObjectsMonitor`.. 264
51. Eventos `TWSApplicationMonitor`. 265
52. Eventos `FileMonitor`. 265
53. Tipos de Ação por Provedor de Ação. 267
54. Critérios de Filtragem de Objetos de Planejamento 282
55. Delimitadores e Caracteres Especiais para o Composer. 283
56. Lista de Comandos do Composer 284
57. Identificadores de Objetos para Cada Tipo de Objeto Definido no Banco de Dados 285
58. Atualização da Definição de Objeto na Exclusão do Objeto Referenciado 286
59. Referential integrity check when deleting an object from the database 286

© Copyright IBM Corp. 1999, 2014

60. Formatos de Saída para Exibir Objetos de Planejamento 297
61. Formatos de Saída para Exibir Objetos de Planejamento 308
62. Objetos extraídos durante o processo de exportação 332
63. Resolvendo o Arquivo de Mapeamento 333
64. Delimitadores e Caracteres Especiais para conman 343
65. Lista de Comandos do Conman 360
66. Alteração de Estado depois do Comando confirm 372
67. Links Abertos 388
68. Estações de Trabalho Iniciadas 439
69. Estações de Trabalho Paradas 444
70. Estações de Trabalho Paradas com stop ;progressive 445
71. Estações de Trabalho Desvinculadas 468
72. Tipos de Tarefa 475
73. Variáveis Tivoli Workload Scheduler suportadas nas definições JSDL. 480
74. Propriedades para tarefas do InfoSphere DataStage 482
75. Propriedades para tarefas sombra 483
76. Propriedades para tarefas do OSLC 483
77. Recursos parcialmente ou não suportados para tarefas no fluxo de tarefas *USERJOBS*. . 491
78. Lista de Comandos Utilitários 493
79. Propriedades adicionais que podem ser utilizadas para definir eventos customizados. . 508
80. Lista de Comandos Utilitários para Estações de Trabalho Dinâmicas 533
81. Formatos de Data 560
82. Lista de Comandos de Relatório 560
83. Programas de Extração de Relatório. 578
84. Campos de Saída de Jbxtract 579
85. Campos de Saída de Prxtract 580
86. Campos de Saída de Caxtract 580
87. Campos de Saída de Paxtract 581
88. Campos de Saída de Rextract 581
89. Campos de Saída de R11xtr 582
90. Campos de Saída de Xdep_job 583
91. Campos de Saída de Xdep_job (continuação) 584
92. Campos de Saída de Xdep_sched 584
93. Campos de Saída de Xfile 585
94. Campos de Saída de Xjob 585
95. Campos de Saída de Xprompts 585
96. Campos de Saída de Xresource 586
97. Campos de Saída de Xsched 586
98. Campos de Saída de Xwhen 586
99. Formatos de Saída de Relatório Suportados 588
100. Resumo dos relatórios de histórico 589
101. Resumo dos relatórios de produção 590
102. Opções de Tarefas de Comandos do Método 604
103. Mensagens da Tarefa LJ (Launch Job) 610
104. Mensagens da Tarefa CF (Check File) 611
105. Mensagens da Tarefa GS (Get Status) 612
106. Dependências da interligação de redes em um ambiente misto 623
107. Transição de Status da Tarefa Sombra 627
108. Critérios de Correspondência para Tarefas Sombra Distribuídas 630
109. Sintaxe da expressão regular. 655
110. Exemplos de expressão regular. 656
111. Estrutura Hierárquica do Aqruivo JSDL 668
112. Tipos e Propriedades de Recursos. 708
113. Comandos Utilizados Contra o Plano 709
114. Comandos de Objetivo Geral 710
115. Comandos do Composer. 716
116. Comandos que Podem Ser Executados a partir de Conman 719
117. Comandos Utilitários Disponíveis para o UNIX e oWindows. 722
118. Comandos Utilitários Disponíveis somente para o UNIX. 723
119. Comandos Utilitários Disponíveis somente para o Windows 724
120. Comandos de relatório 724
121. Programas de Extração de Relatório 725
122. Parâmetros de entrada para o cenário de tarefa de ramificação negativa 742
123. Parâmetros de entrada para o cenário da tarefa de padrão 745
124. Parâmetros de entrada para o cenário de tarefa de padrão negado 747
125. Parâmetros de entrada para o padrão no cenário de linha de padrão 749
126. Parâmetros de entrada para o cenário de padrão negado na linha de padrão 751
127. Parâmetros de entrada para o cenário de comparação Numérica 754
128. Parâmetros de entrada para o cenário de condição complexa. 756
129. Parâmetros de entrada para o cenário de pausa e liberação 760
130. Parâmetros de entrada para o cenário de ação de sinal 765
131. Parâmetros e Valores 773
132. Descrição de operadores aritméticos 774

Sobre esta publicação

O IBM® Tivoli Workload Scheduler simplifica o gerenciamento de sistemas por meio de ambientes distribuídos, integrando as funções de gerenciamento de sistemas. O Tivoli Workload Scheduler planeja, automatiza e controla o processamento de toda a carga de trabalho de produção de seu empreendimento. O *Guia do Usuário e Referência do Tivoli Workload Scheduler* fornece informações detalhadas sobre a interface da linha de comandos, a linguagem de planejamento e comandos utilitários para o Tivoli Workload Scheduler.

O que Há de Novo nesta Liberação

Saiba o que há de novo nesta liberação.

Para obter informações sobre as funções novas ou alteradas nesta liberação, consulte *Tivoli Workload Automation: Overview*, seção *Resumo dos Aprimoramentos*.

Para obter informações sobre os APARs abordados nesta liberação, consulte o Tivoli Workload Scheduler Notas sobre a Liberação em http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27041032 e o Dynamic Workload Console Notas sobre a Liberação em http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27041033.

O Que Há de Novo nesta Publicação

Saiba o que há de novo nesta publicação.

As seguintes informações foram incluídas ou alteradas nesta publicação:

•

Quem Deve Ler Esta Publicação

Saiba sobre o público desta publicação.

Essa publicação é destinada às pessoas envolvidas no planejamento, programação, monitoramento ou gerenciamento de um ambiente de planejamento de carga de trabalho. Normalmente essas pessoas são operadores e administradores do Tivoli Workload Scheduler.

Publicações

O produto Tivoli Workload Automation é suportado por um conjunto de publicações.

Para obter uma lista de publicações na biblioteca do produto Tivoli Workload Automation, consulte *Publicações* em *Referência* na documentação do produto.

Para obter uma lista de termos utilizados no produto Tivoli Workload Automation, consulte *Glossário* em *Referência* na documentação do produto.

© Copyright IBM Corp. 1999, 2014

Acessibilidade

Os recursos de acessibilidade ajudam usuários com deficiências físicas, como mobilidade restrita ou visão limitada, a usar os produtos de software com êxito.

Com este produto, você pode utilizar tecnologias de assistência para escutar e navegar na interface. Também pode utilizar o teclado em vez do mouse para operar todos os recursos da GUI (interface gráfica com o usuário).

Para obter informações integrais a respeito do Dynamic Workload Console, consulte o Apêndice de Acessibilidade no *Guia do Usuário e Referência do IBM Tivoli Workload Scheduler*.

Treinamento Técnico Tivoli

O Tivoli fornece treinamento técnico.

Para obter informações sobre treinamento técnico Tivoli, consulte o seguinte Web site IBM Tivoli Education:

http://www.ibm.com/software/tivoli/education

Informações sobre suporte

A IBM fornece várias maneiras de obter suporte ao encontrar um problema.

Se ocorrer um problema com seu software IBM, você quer resolvê-lo rapidamente. A IBM oferece as seguintes maneiras para você obter o suporte necessário:

- Procurando nas bases de conhecimento: É possível procurar em uma grande coleção de problemas conhecidos e soluções alternativas, Technotes e outras informações.
- Obtendo correções: É possível localizar as correções mais recentes que já estão disponíveis para seu produto.
- Entrando em contato com o Suporte de Software IBM: Se ainda não conseguir resolver seu problema, e precisar trabalhar com alguém da IBM, será possível utilizar as várias maneiras de entrar em contato com o Suporte de Software IBM.

Para obter informações adicionais sobre estas três maneiras de resolver problemas, consulte o apêndice sobre informações de suporte no *Tivoli Workload Scheduler: Guia de Resolução de Problemas*.

Convenções Utilizadas Nesta Publicação

Saiba quais convenções são utilizadas nesta publicação.

Esta publicação utiliza várias convenções para termos e ações especiais, comandos e caminhos dependentes do sistema operacional, sintaxe de comando e gráficos de margem.

Convenções de Fontes

Esta publicação utiliza as seguintes convenções de tipo de caractere:

Negrito

- Comandos em minúsculas e comandos que misturem letras minúsculas e maiúsculas que possam ser difíceis de serem distinguidos do texto ao redor
- Controles de interface (caixas de opções, botões de comando, botões de opções, botões de giro, campos, pastas, ícones, quadros de listagem, itens dentro de quadros de listagem, listas com várias colunas, contêineres, opções de menu, nomes de menu, guias, folhas de propriedade), rótulos (como **Dica:** e **Considerações sobre o sistema operacional**:)
- Palavras-chave e parâmetros em um texto

Itálico

- Palavras definidas no texto
- Ênfase de palavras (palavras como palavras)
- Novos termos no texto (exceto em uma lista de definições)
- Variáveis e valores que você deve fornecer

`Espaçamento fixo`

- Exemplos e exemplos de código
- Nomes de arquivos, palavras-chave de programação e outros elementos que são difíceis de serem distinguidos do texto ao redor
- Texto de mensagem e prompts dirigidos ao usuário
- Texto que o usuário deve digitar
- Valores para argumentos ou opções de comando

variáveis e caminhos dependentes do sistema operacional

Esta publicação utiliza a convenção do UNIX para especificar variáveis de ambiente e para notação de diretórios, exceto onde o contexto ou o caminho de exemplo seja especificamente Windows.

Ao utilizar a linha de comandos do Windows, substitua **$***variable* por **%** *variable***%** para as variáveis de ambiente e substitua cada barra (**/**) por uma barra invertida (\) nos caminhos do diretório. Os nomes das variáveis de ambiente nem sempre são os mesmos nos ambientes do Windows e UNIX. Por exemplo, %TEMP% no Windows é equivalente a $tmp nos ambientes UNIX.

Nota: Se você estiver utilizando o shell bash em um sistema Windows, será possível utilizar as convenções UNIX.

Sintaxe do Comando

Esta publicação utiliza a seguinte sintaxe em todos os lugares onde os comandos são descritos:

Tabela 1. Sintaxe do Comando

Convenção de sintaxe	Descrição	Exemplo
Nome do comando	A primeira palavra ou conjunto de caracteres consecutivos.	**conman**
Colchetes ([ ])	As informações entre colchetes ([ ]) são opcionais. Tudo o que estiver fora dos colchetes deve ser especificado.	[**-file** *definition_file*]
Chaves ({ })	As chaves ({ }) identificam um conjunto de opções exclusivas mutuamente, quando uma opção é exigida.	{**-prompts** | **-prompt** *prompt_name* }
Sublinhado (_)	Um sinal de sublinhado (_) conecta várias palavras em uma variável.	*prompt_name*

Tabela 1. Sintaxe do Comando (continuação)

Convenção de sintaxe	**Descrição**	**Exemplo**
Barra vertical (|)	As opções exclusivas mutuamente são separadas por uma barra vertical (|). É possível digitar uma das opções separadas por uma barra vertical, mas não poderá digitar várias opções em uma única utilização do comando.	{**-prompts** | **-prompt** *prompt_name* }
Negrito	O texto em **Negrito** designa informações de literal que devem ser digitadas na linha de comando exatamente como mostrado. Isso se aplica aos nomes de comandos e às opções não variáveis.	**composer add** *file_name*
Itálico	O texto em Itálico é uma variável e deve ser substituída pelo seu valor real. No exemplo à direita, o usuário substituiria file_name pelo nome do arquivo específico.	*file_name*
Reticências (...)	As reticências (...) indicam que a opção anterior pode ser repetida várias vezes com diferentes valores. Elas podem ser utilizadas dentro ou fora de parênteses.	[**–x** *file_name*]... As reticências fora de colchetes indicam que **–x** *file_name* é opcional e pode ser repetido da seguinte forma: *–x file_name1 –x file_name2–x file_name3* [**–x** *file_name*...] As reticências dentro de colchetes indicam que **–x** *file_name* é opcional e a variável do arquivo pode ser repetida da seguinte forma: *–x file_name1 file_name2 file_name3* **–x** *file_name* [**–x** *file_name*]... As reticências utilizadas com esta sintaxe indicam que é necessário especificar **–x** *file_name* pelo menos uma vez.

Capítulo 1. Visão Geral do Tivoli Workload Scheduler

O IBM Tivoli Workload Scheduler fornece a capacidade de gerenciar seu ambiente de produção e automatizar diversas atividades do operador. O Tivoli Workload Scheduler gerencia o processamento de tarefas, resolve interdependências e ativa e monitora tarefas. Como as tarefas são iniciadas assim que suas dependências são atendidas, o tempo inativo é minimizado e o rendimento do processamento é aprimorado de forma significativa. Se uma tarefa falhar, o Tivoli Workload Scheduler gerencia o processo de recuperação com pouca ou nenhuma intervenção do operador.

Este capítulo está dividido nas seções a seguir:

- "Entendendo Conceitos Básicos"
- "Interfaces com o Usuário do Tivoli Workload Scheduler" na página 26
- "Iniciando a Produção" na página 28

Entendendo Conceitos Básicos

Esta seção descreve os conceitos básicos do Tivoli Workload Scheduler e está dividida nas seguintes seções:

- "Objetos de Banco de Dados do Tivoli Workload Scheduler"
- "A Rede do Tivoli Workload Scheduler" na página 19
- "Configurando seu Ambiente de Tempo de Execução do Tivoli Workload Scheduler" na página 20
- "Definindo Atividades de Planejamento Usando o Tivoli Workload Scheduler" na página 21
- "Gerenciando Atividades de Planejamento de Produção com o Tivoli Workload Scheduler" na página 25

Objetos de Banco de Dados do Tivoli Workload Scheduler

Esta seção apresenta os objetos de banco de dados do Tivoli Workload Scheduler com os quais você trabalha. Os seguintes objetos de banco de dados são descritos:

- Tarefa, consulte "Tarefa" na página 2
- Fluxo de tarefas, consulte "Fluxo de Tarefas" na página 2
- Aplicativo de carga de trabalho, consulte "Aplicativo de carga de trabalho" na página 3
- Ciclo de execução, consulte "Ciclo de Execução" na página 4
- Grupo de ciclos de execução, consulte "Grupo do Ciclo de Execução" na página 5
- Calendário, consulte "Calendário" na página 10
- Prompt, consulte "Aviso" na página 10
- Estação de trabalho, consulte "Estação de Trabalho" na página 10
- Classe de estação de trabalho, consulte "Classe de Estação de Trabalho" na página 14
- Domínio, consulte "Domínio" na página 15
- Regra de evento, consulte "Regra de Evento" na página 18
- Recurso, consulte "Resource" na página 18

© Copyright IBM Corp. 1999, 2014

- Parâmetro, consulte “Parâmetro” na página 18
- Tabela de variáveis, consulte “Tabela de Variáveis” na página 19

Tarefa

Uma *tarefa* é uma unidade de trabalho que especifica uma ação, tal como um backup de dados semanal, a ser desempenhada em estações de trabalho específicas na rede do Tivoli Workload Scheduler. Em um ambiente distribuído do Tivoli Workload Scheduler, as tarefas podem ser definidas independentemente de fluxos de tarefas ou dentro de uma definição de fluxo de tarefas.

Os tipos de tarefas podem ser divididos entre tarefas do Tivoli Workload Scheduler existentes e tipos de tarefas com opções avançadas. Os tipos de tarefas existentes são scripts ou comandos genéricos que é possível customizar de acordo com suas necessidades. Os tipos de tarefas com opções avançadas são tarefas designadas para executar operações específicas, como operações de banco de dados, de transferência de arquivos, Java e de serviço da Web. Esses tipos de tarefas são planejados somente em agentes, conjuntos e conjuntos dinâmicos.

Se quiser usar o recurso dinâmico ao planejar os tipos de tarefas com opções avançadas, você planeja eles em conjuntos e conjuntos dinâmicos, que designam a tarefa dinamicamente ao melhor recurso disponível. Se estiver interessado somente na definição dos tipos de tarefas com opções avançadas, sem usar o recurso de planejamento dinâmico, você planeja essas tarefas em um agent específico, no qual a tarefa é executada estaticamente.

Independentemente de se o mecanismo do Tivoli Workload Scheduler é distribuído ou baseado no z/OS, é possível definir localmente uma *tarefa sombra* para mapear uma instância de tarefa remota em um mecanismo diferente do Tivoli Workload Scheduler.

Para obter informações sobre como definir tarefas, consulte “Definição de tarefa” na página 156.

Para obter informações sobre como definir estações de trabalho, consulte “Definição de estação de trabalho” na página 139.

Fluxo de Tarefas

Um *fluxo de tarefa* é uma seqüência de tarefas a serem executadas, juntamente com dependências de horas, prioridades e outras dependências que determinam a ordem de processamento. Cada fluxo de tarefas é designado a um horário para ser executado, representado pelo ciclo de execução com tipo calendário, conjunto de datas ou taxas de repetição.

Dependências em um ambiente distribuído:

Você pode ter dependências entre as tarefas e os fluxos de tarefas: Podem ser:

Dependências Internas

Essas são as dependências estabelecidas entre as tarefas que pertencem ao mesmo fluxo de tarefas.

Dependências externas

Essas são as dependências entre os fluxos de tarefas ou entre os fluxos de tarefas e as tarefas que pertencem a outros fluxos de tarefas ou entre as tarefas que pertencem a diferentes fluxos de tarefas.

Dependências da Interligação de Redes
Estas são dependências em tarefas ou fluxos de tarefas que são executados em outra rede Tivoli Workload Scheduler. As dependências da interligação de redes requerem que uma estação de trabalho do agente de rede se comunique com a rede externa do Tivoli Workload Scheduler.

As dependências em recursos são suportadas pelo Tivoli Workload Scheduler, ambas nos ambientes distribuídos e z/OS.

Para obter informações sobre como definir fluxos de tarefa, consulte "Definição de fluxo de tarefas" na página 216.

Aplicativo de carga de trabalho

Um aplicativo de carga de trabalho é um ou mais fluxos de tarefas juntamente com todas as tarefas referenciadas que podem ser compartilhadas com outros ambientes do Tivoli Workload Scheduler por meio de um processo de implementação fácil.

Um aplicativo de carga de trabalho é um objeto de banco de dados do Tivoli Workload Scheduler que age como um contêiner para um ou mais fluxos de tarefas. É possível usar oaplicativos de carga de trabalho para padronizar uma solução de automação de carga de trabalho para que a solução possa ser reutilizada em um ou mais ambientes do Tivoli Workload Scheduler automatizando, portanto, os processos de negócios.

É possível preparar um modelo do aplicativo de carga de trabalho em um ambiente de origem do Tivoli Workload Scheduler e, em seguida, exportá-lo para que ele possa ser implementado em um ambiente de destino. O processo de exportação extrai do ambiente de origem de todos os elementos necessários para reproduzir a solução em outro ambiente. Ele produz um arquivo compactado que contém diversos arquivos necessários para importar oaplicativo de carga de trabalho para o ambiente de destino. Estes arquivos contêm uma definição dos objetos no ambiente de origem extraído do banco de dados do Tivoli Workload Scheduler. Para os elementos que dependem da topologia do ambiente de destino, é necessária alguma configuração manual. Por exemplo, as definições extraídas do ambiente de origem contêm referências a estações de trabalho que não existem no ambiente de destino. Por isso, antes de prosseguir com a importação, deve ser feito um mapeamento de alguns dos elementos, associando o nome do objeto no ambiente de destino.

O modelo do aplicativo de carga de trabalho exportado contém definições ou referências para todos os seguintes objetos:

- Fluxos de Tarefas
- Tarefas
- Estações de trabalho, classes de estações de trabalho
- Calendários
- Prompts
- Ciclos de Execução
- Grupos de ciclos de execução
- Recursos
- Dependências da Interligação de Redes
- Dependências externas

Para obter informações sobre como definir modelos do aplicativo de carga de trabalho, consulte “Definindo os aplicativos de carga de trabalho” na página 271.

Ciclo de Execução

Um ***ciclo de execução*** especifica os dias em que um fluxo de tarefas está planejado para execução. Um ciclo é definido para um fluxo de tarefa específico e não pode ser utilizado por vários fluxos de tarefa. É possível especifcar os seguintes tipos de ciclo de execução:

simples
Um conjunto específico de dias, definidos pelo usuário, que um fluxo de tarefas é executado. Um ciclo de execução simples é definido para um fluxo de tarefas específico e não pode ser utilizado por outros fluxos de tarefas.

diário Um ciclo de execução que especifica se o fluxo de tarefas é executado de acordo com uma freqüência em dias e com o tipo que você definiu. Por exemplo, ele pode ser diário, a cada três dias ou apenas em dias úteis.

semanal
Um ciclo de execução que especifica os dias da semana que um fluxo de tarefas é executado. Por exemplo, um fluxo de tarefas pode ser especificado para executar toda segunda-feira, quarta-feira e sexta-feira utilizando um ciclo de execução semanal.

mensalmente
Um ciclo de execução que especifica se o fluxo de tarefas é executado de acordo com um dia do mês ou uma data que você definiu. Por exemplo, ele pode ser executado todo primeiro e segundo dias do mês, a cada dois meses, toda primeira segunda-feira e segunda terça-feira do mês, a cada três meses.

anualmente
Um ciclo de execução que especifica se um fluxo de tarefas é executado, por exemplo, anualmente ou cada três anos.

baseado em deslocamento
Um ciclo de execução utiliza uma combinação de períodos e deslocamentos definidos pelo usuário. Por exemplo, um deslocamento de 3 em um período de 15 dias Ú o terceiro dia a partir do início do período. É mais prático utilizar ciclos de execução baseados em deslocamento quando o ciclo é baseado em períodos cíclicos. O termo só é usado como tal no Tivoli Workload Scheduler for z/OS, mas o conceito se aplica também ao produto distribuído.

baseado em regra
Um ciclo de execução que utiliza regras baseadas em listas de números ordinais, tipos de dias e intervalos de calendário comuns (ou nomes de períodos no Tivoli Workload Scheduler for z/OS). Por exemplo, a última quinta-feira de cada mês. Os ciclos de execução baseados em regras são baseados em períodos convencionais, como meses do calendário, semanas do ano e dias da semana. No Tivoli Workload Scheduler for z/OS, os ciclos de execução também podem ser baseados em períodos que são definidos, como por exemplo, um semestre. O termo só é usado como tal no Tivoli Workload Scheduler for z/OS, mas o conceito se aplica também ao produto distribuído. Você também pode especificar uma regra que estabeleça quando um fluxo de tarefas será executado se ele cair em um dia livre.

Qualquer um destes tipos de ciclo de execução podem ser inclusivos ou exclusivos; ou seja:

inclusivo
Um ciclo de execução que especifica os dias e horas que um fluxo de tarefas é planejado para ser executado. Ciclos de execução inclusivos fornecem precedência para ciclos de execução inclusivos.

exclusivo
O ciclo de execução que especifica dias e horas em que um fluxo de tarefas não pode ser executado. Os ciclos de execução exclusivos têm precedência sobre os ciclos de execução inclusivos.

Grupo do Ciclo de Execução

Opcionalmente, é possível definir um grupo do ciclo de execução para seu fluxo de tarefas, em vez de ou além de diversos ciclos de execução únicos.

Um grupo do ciclo de execução é uma lista de ciclos de execução que são combinados para produzir um conjunto de datas de execução.

Usando grupos de ciclos de execução, é possível beneficiar-se das seguintes vantagens:

Um grupo do ciclo de execução é um objeto de banco de dados distinto
Ele é definido por si próprio e pode ser correspondido com um ou mais fluxos de tarefas. Ele não é definido como parte de um fluxo de tarefas específico, como ciclos de execução únicos.

O mesmo grupo do ciclo de execução pode ser usado em diferentes fluxos de tarefas
Isso melhora a usabilidade geral dos ciclos de execução, porque é possível especificar o mesmo grupo do ciclo de execução em diversos fluxos de tarefas, evitando a necessidade de ter diversas definições de ciclos de execução para as mesmas regras de planejamento.

Os grupos de ciclos de execução aprimoram o uso de ciclos de execução exclusivos
Os ciclos de execução exclusivos (ou negativos) são usados para gerar ocorrências negativas, que identificam os dias em que um fluxo de tarefas normalmente seria planejado, mas não é necessário. A soma dos ciclos de execução exclusivos é subtraída dos ciclos de execução inclusivos. Uma ocorrência negativa sempre cancela quaisquer ocorrências positivas correspondentes e é possível especificar uma ocorrência negativa apenas se o equivalente positivo já existir. Uma correspondência exata de dias, bem como as restrições de tempo, é necessária entre os ciclos de execução exclusivos e inclusivos para o cancelamento ocorrer. Os grupos de ciclos de execução incluem muita flexibilidade, permitindo que os usuários apliquem ciclos de execução a um subconjunto dos positivos em vez de aplicar a todos os eles. Agrupe seus ciclos de execução em *subconjuntos* para que os ciclos de execução exclusivos possam ser aplicados apenas às ocorrências positivas geradas pelos ciclos de execução pertencentes ao mesmo conjunto.

Os ciclos de execução devem ser organizados em *subconjuntos* em um grupo do ciclo de execução. Os subconjuntos estão sempre em um relacionamento**OR** lógico uns com os outros. O resultado do grupo do ciclo de execução é sempre uma data ou um conjunto de datas; ele não pode ser negativo.

Por exemplo, talvez você queira que seu fluxo de tarefas seja executado todos os dias do mês, exceto no último dia do mês. Mas, você também

deseja que ele seja planejado no último dia do ano (o último dia de dezembro). É possível definir um grupo do ciclo de execução usando subconjuntos, da seguinte forma:

Subconjunto 1

- **Ciclo de execução 1** - Ciclo de execução inclusivo todos os dias do mês
- **Ciclo de execução 2** - Ciclo de execução exclusivo no último dia do mês

Subconjunto 2

- **Ciclo de execução 3** - Ciclo de execução inclusivo em 31 de dezembro

em que, o ciclo de execução 2 cancela o último dia de cada mês no Subconjunto 1, enquanto o ciclo de execução 3 gera 31 de dezembro como uma data separada e, portanto, é possível planejar o fluxo de tarefas no dia 31 de dezembro.

Os grupos de ciclos de execução permitem o uso de um AND lógico entre ciclos de execução individuais no subconjunto

Por padrão, os ciclos de execução em um subconjunto estão em um relacionamento**OR** lógico, mas é possível alterar isso para um **AND** lógico, se o resultado do grupo do ciclo de execução for uma data positiva ou um conjunto de datas (Inclusivo). Para cada ciclo de execução, é possível especificar qualquer operador (**AND** ,**OR**), obtendo o seguinte comportamento:

1. Todos os ciclos de execução do grupo que estão no relacionamento*AND* são calculados primeiro. O resultado desse cálculo é uma data ou um conjunto de datas.
2. Em seguida, todos os ciclos de execução em um relacionamento *OR* são incluídos no resultado da etapa anterior.

Um comportamento semelhante é aplicado a ciclos de execução inclusivos e exclusivos para determinar a data final ou o conjunto de datas de um grupo.

Inclusivo (A)
Ciclo de execução baseado em regras. Selecione os dias em que o fluxo de tarefas deverá ser executado, se pertencer a todos os tipos A do conjunto de ciclos de execução.

Exclusivo (D)
Ciclo de execução baseado em regras de exclusão. Selecione os dias em que o fluxo de tarefas NÃO deverá ser executado, se pertencer a todos os tipos D do conjunto de ciclos de execução.

Por exemplo, é possível incluir duas condições juntas:

```
Executar na quarta-feira "AND" no 8º dia útil do mês.
```

Dessa forma, as únicas datas planejadas serão qualquer 8º dia útil do mês que caia em uma quarta-feira.

Compatibilidade total com ciclos de execução *tradicionais*

Os ciclos de execução *tradicionais* especificados na definição de fluxo de tarefas podem referenciar grupos de ciclos de execução, com a possibilidade de especificar turnos ou deslocamentos neles (assim como períodos for z/OS ou calendários para sistemas distribuídos).

Um conjunto de datas (inícios de intervalos) é criado automaticamente em um nível de ciclo de execução diretamente (de forma inclusiva ou exclusiva) com deslocamentos, ou na regra. Este é um processo de duas etapas com ciclos de execução:

1. Defina o "evento de negócios" chave, por exemplo, Final do Mês, usando ciclos de execução e regras de tempo livre
2. Defina regras que usam as datas do "evento de negócios" como os intervalos em relação aos quais a outra execução em lote pode ser planejada.

Por exemplo, você tem um *Processo de Final do Mês* que é executado na Última Sexta-feira de um mês, mas que se muda para o próximo dia útil, exceto em dezembro, quando ele é executado na terceira sexta-feira do mês. Esta regra de planejamento pode ser definida com algumas regras, ciclos de execução e regras de tempo livre.

Dois dias úteis antes do Final do Mês é necessário executar um processo de pré-validação para permitir que os problemas sejam tratados antes da execução. Não é possível escolher a última quarta-feira do mês, porque, em alguns meses, isso pode ocorrer após a última sexta-feira. De forma semelhante, se a última sexta-feira foi um tempo livre, a última quarta-feira não terá 2 dias úteis anteriores a ela, porque a regra do Tempo Livre se aplica APENAS ao dia em que a regra está contida, ela não pode verificar mais nada.

Talvez muitas outras execuções em lote também precisem ser executadas em um determinado número de dias antes ou após o Final do Mês, mas as mesmas restrições se aplicam.

Agora é possível definir o trabalho a ser executado em relação a algo definido por uma combinação de ciclos de execução e de regras de tempo livre.

Uso de calendários com ciclos de execução em um grupo do ciclo de execução

Opcionalmente, é possível especificar mais de um calendário para calcular a definição de dias úteis e de folga para um ciclo de execução. O calendário primário é usado para calcular quais dias úteis são válidos e um calendário secundário é usado para calcular datas de folga específicas. Se as datas calculadas de acordo com o calendário secundário corresponderem aos dias úteis no calendário primário, a tarefa será planejada; se elas não corresponderem, a tarefa não será planejada.

Por exemplo, uma empresa global que executa carga de trabalho nos Estados Unidos para muitos outros países precisa de muitas combinações de calendários para assegurar que as tarefas em lote sejam executadas apenas em um dia que seja um dia útil nos Estados Unidos e no outro país. O calendário pode ser definido no nível do fluxo de tarefas e, se não for especificado, será usado um calendário padrão. No entanto, o calendário no nível de ciclo de execução, sempre que definido, pode ser usado como o calendário secundário e o calendário do fluxo de tarefas (ou padrão) pode ser usado como o calendário primário.

Por exemplo, o calendário Primário pode ser *WORKDAYS*, que é definido como MONDAY a FRIDAY, excluindo dias de feriados nos EUA. Também pode ser necessário calcular as execuções de tarefas com base no calendário*HKWORK*, que é definido como segunda a sexta-feira, excluindo datas de feriados em Hong Kong. A tarefa pode ter vários planejamentos:

- Execução em dias úteis, mas não no último dia útil e não às segundas-feiras
- Execução às segundas-feiras, mas não no último dia útil
- Execução no último dia útil

Como cada planejamento é calculado em relação ao calendário*WORKHK*, ele também é verificado em relação ao calendário*WORKDAYS* para assegurar que será planejado em um dia útil nos EUA.

O uso de restrições de tempo com grupos de ciclos de execução

É possível especificar restrições de tempo para definir quando o processamento deve iniciar ou o tempo após o qual o processamento não deve mais iniciar. Para isso, é possível associar *restrições de tempo* à tarefa, fluxos de tarefas, ciclos de execução e grupos de ciclos de execução. Ao definir uma restrição de tempo, basicamente você obtém um *tempo*. Como é possível associar restrições de tempo a diversos objetos, a seguinte hierarquia mostra a ordem na qual as diferentes restrições de tempo são consideradas para realmente definir quando iniciar o processamento:

1. Restrição de tempo definida no ciclo de execução para o fluxo de tarefas
2. Restrição de tempo definida no fluxo de tarefas
3. Restrição de tempo definida no ciclo de execução contido no grupo do ciclo de execução associado ao fluxo de tarefas.
4. Restrição de tempo definida no grupo do ciclo de execução associado ao fluxo de tarefas.
5. Início do Dia

Isso significa que:

Restrições de tempo no fluxo de tarefas

Substituir e ter precedência *sobre quaisquer outras restrições de tempo* definidas nos ciclos de execução ou grupos de ciclos de execução associados ao fluxo de tarefas.

Não há restrições de tempo no fluxo de tarefas nem no grupo do ciclo de execução

O grupo gera apenas uma data que é o*Início do Dia*. Se deslocamentos e regras de tempo livre tiverem que ser calculados, o cálculo sempre começará a partir do *Início do Dia*.

Restrições de tempo no grupo do ciclo de execução (não no fluxo de tarefas)

Restrições de tempo (e possível deslocamento) são calculadas começando a partir do *Início do Dia* e a data e hora resultantes indicam o início do processamento.

Exemplos

Tabela 2. Cenário 1. Não há restrição de tempo no grupo do ciclo de execução

Grupo do Ciclo de Execução	**Data planejada**	**Início Mais Antecipado**
Grupo do Ciclo de Execução	24/10	24/10
Grupo de ciclos de execução com deslocamento (+ 3 dias)	27/10 (Sábado)	27/10/ (Sábado)
Grupo de ciclos de execução com regra de tempo livre	29/10/ (Segunda-feira)	29/10/ (Segunda-feira)
Ciclo de execução no fluxo de tarefas com restrições de tempo		
Ciclo de execução no fluxo de tarefas com turno de + 4 dias úteis	02/11 (Sexta-feira)	02/11 (Sexta-feira)
Ciclo de execução no fluxo de tarefas com regra de tempo livre	02/11 (Sexta-feira)	02/11 (Sexta-feira)

Tabela 2. Cenário 1. Não há restrição de tempo no grupo do ciclo de execução (continuação)

Grupo do Ciclo de Execução	Data planejada	Início Mais Antecipado
Ciclo de execução no fluxo de tarefas com início mais antecipado +1 13h	02/11 (Sexta-feira)	03/11 (Sábado) 13h
Ciclo de execução no fluxo de tarefas sem restrições de tempo		
Ciclo de execução no fluxo de tarefas com turno de + 4 dias úteis	02/11 (Sexta-feira)	02/11 (Sexta-feira) Início do Dia
Ciclo de execução no fluxo de tarefas com regra de tempo livre	02/11 (Sexta-feira)	02/11 (Sexta-feira) Início do Dia

Tabela 3. Cenário 2. Restrição de tempo no grupo do ciclo de execução sem deslocamento

Grupo do Ciclo de Execução	Data planejada	Início Mais Antecipado
Grupo do Ciclo de Execução	24/10	24/10
Grupo de ciclos de execução com deslocamento de calendário (+ 3 dias)	27/10/ (Sábado)	27/10/ (Sábado)
Grupo de ciclos de execução com regra de tempo livre	29/10/ (Segunda-feira)	29/10/ (Segunda-feira)
Ciclo de execução no fluxo de tarefas com restrições de tempo		
Ciclo de execução no fluxo de tarefas com turno de + 4 dias úteis	02/11 (Sexta-feira)	02/11 (Sexta-feira)
Ciclo de execução no fluxo de tarefas com regra de tempo livre	02/11 (Sexta-feira)	02/11 (Sexta-feira)
Ciclo de execução no fluxo de tarefas com início mais antecipado +1 13h	02/11 (Sexta-feira)	03/11 (Sábado) 13h
Ciclo de execução no fluxo de tarefas sem restrições de tempo		
Ciclo de execução no fluxo de tarefas com turno de + 4 dias úteis	02/11 (Sexta-feira)	02/11 (Sexta-feira) Início do Dia
Ciclo de execução no fluxo de tarefas com regra de tempo livre	02/11 (Sexta-feira)	02/11 (Sexta-feira) Início do Dia

Tabela 4. Cenário 3. Restrição de tempo no grupo do ciclo de execução com deslocamento (+1 12:00)

Grupo do Ciclo de Execução	Data planejada	Início Mais Antecipado
Grupo do Ciclo de Execução	24/10	24/10
Grupo de ciclos de execução com deslocamento de calendário (+ 3 dias)	27/10/ (Sábado)	27/10/ (Sábado)
Grupo de ciclos de execução com regra de tempo livre	29/10/ (Segunda-feira)	29/10/ (Segunda-feira)
Grupo de ciclos de execução com deslocamento +1 12:00	29/10/ (Segunda-feira)	30/10 12:00 (Terça-feira)
Ciclo de execução no fluxo de tarefas com restrições de tempo		
Ciclo de execução no fluxo de tarefas com turno de + 4 dias úteis	02/11 (Sexta-feira)	02/11 (Sexta-feira)
Ciclo de execução no fluxo de tarefas com regra de tempo livre	02/11 (Sexta-feira)	02/11 (Sexta-feira)
Ciclo de execução no fluxo de tarefas com início mais antecipado +1 13h	02/11 (Sexta-feira)	03/11 (Sábado) 13h
Ciclo de execução no fluxo de tarefas sem restrições de tempo		
Ciclo de execução no fluxo de tarefas com turno de + 4 dias úteis	02/11 (Sexta-feira)	11/03 12:00 (Sábado)
Ciclo de execução no fluxo de tarefas com regra de tempo livre	02/11 (Sexta-feira)	11/03 12:00 (Sábado)

Disponibilidade do comando GENDAYS no nível do grupo do ciclo de execução

Usando GENDAYS, é possível verificar o resultado da combinação de todos os ciclos de execução no grupo.

Calendário

Um *calendário* é uma lista de datas que definem se e quando um fluxo de tarefas é executado.

Um calendário também pode ser designado para utilização como um calendário de *dias livres* em um fluxo de tarefas. Um calendário de dias livres é aquele designado a um fluxo de tarefas para representar os dias quando o fluxo de tarefas e suas tarefas não são executados. Ele também pode ser utilizado para designar sábados ou domingos, ou ambos, como dias úteis. Por convenção, muitos usuários definem um calendário de dias livres denominado *holidays*, em que habitualmente sábado e domingo são especificados como dias livres.

Para obter informações sobre como definir calendários, consulte "Definição do calendário" na página 197.

Aviso

Um *aviso* identifica uma mensagem textual que é exibida para o operador e pára o processamento da tarefa ou do fluxo de tarefas até que uma resposta afirmativa seja recebida (ou manualmente do operador ou automaticamente por uma ação da regra de evento). Após o aviso ser respondido, o processamento continua. Os prompts podem ser utilizados como dependências em jobs e fluxos de job. Você também pode utilizar avisos para alertar um operador de uma tarefa específica que foi executada. Nesse caso, a resposta do operador não é necessária.

Existem três tipos de avisos:

global ou nomeado
: Um aviso que é definido no banco de dados como um objeto de planejamento. Ele é identificado por um nome exclusivo e pode ser utilizado por qualquer tarefa ou fluxo de tarefas.

local ou ad-hoc
: Esse tipo de aviso é definido dentro de uma definição de tarefa ou fluxo de tarefas. Ele não tem nome e não está definido como um objeto de planejamento no banco de dados; portanto, não pode ser utilizado por outras tarefas ou fluxos de tarefas.

recuperação ou encerramento de forma anormal
: Um tipo especial de aviso que você define para ser utilizado quando uma tarefa é encerrada de forma anormal. A resposta para esse aviso determina o resultado da tarefa ou do fluxo de tarefas aos quais ele pertence. Um aviso de recuperação também pode ser associado a uma ação e a um tipo especial de tarefa chamado*tarefa de recuperação*.

Para obter informações sobre como definir prompts, consulte "Definição do Prompt" na página 204.

Estação de Trabalho

Esta seção fornece informações sobre o uso de estações de trabalho para o planejamento de tarefas e fluxos de tarefas. Se, em vez disso, você desejar aprender sobre estações de trabalho porque está planejando sua rede, será possível localizar as informações necessárias no *Guia de Planejamento e Instalação do Tivoli Workload Scheduler*.

O sistema de computador no qual você executa suas tarefas e fluxos de tarefas é chamado de *estação de trabalho*. Ao definir uma tarefa ou fluxo de tarefas no banco de dados do Tivoli Workload Scheduler, você identifica as definições da estação de

trabalho para os sistemas de computadores físicos ou virtuais nos quais sua tarefa está planejada para ser executada. Estações de trabalho podem ser agrupadas logicamente em *classes da estação de trabalho* e organizadas hierarquicamente em *domínios*, gerenciadas por *gerenciadores de domínio*.

Para obter mais informações sobre classes da estação de trabalho, consulte "Classe de Estação de Trabalho" na página 14, e para domínios, consulte "Domínio" na página 15.

Ao criar uma definição de estação de trabalho para um sistema em sua rede, você define um conjunto de características que identificam exclusivamente o sistema e afetam a maneira como as tarefas são executadas nele. Alguns exemplos destas características são o endereço IP da estação de trabalho, se ela estiver posicionada atrás de um firewall, a comunicação segura ou não segura, o fuso horário no qual a estação de trabalho está localizada e a identidade de seu gerenciador de domínio.

As estações de trabalho na rede de planejamento do Tivoli Workload Scheduler executam a tarefa e o processamento do fluxo de tarefas, mas também podem ter outras funções. Quando sua rede foi projetada, essas funções foram designadas a essas estações de trabalho para atender às necessidades específicas de seus negócios. A lista a seguir descreve todas as funções da estação de trabalho:

Gerenciador de domínio mestre
Uma estação de trabalho atuando como hub de gerenciamento para a rede. Ela gerencia todos os seus objetos de planejamento. Essa estação de trabalho é registrada no banco de dados Tivoli Workload Scheduler como **principal**.

Backup do Gerenciador de Domínio Principal
Uma estação de trabalho que pode atuar como backup para o gerenciador de domínio principal quando ocorrem problemas. Ela é efetivamente um gerenciador de domínio principal, aguardando para ser ativado. Seu uso é opcional. Saiba mais sobre a comutação para um gerenciador de domínio principal de backup no *Tivoli Workload Scheduler Administration Guide*. Esta estação de trabalho deve ser instalada como "gerenciador de domínio principal configurado como backup". Essa estação de trabalho é registrada no banco de dados Tivoli Workload Scheduler como **fta**.

Gerenciador de domínio
Uma estação de trabalho que controla um domínio e compartilha responsabilidades de gerenciamento para parte da rede do Tivoli Workload Scheduler. Ela é instalada como um agente e, em seguida, configurada como uma estação de trabalho do gerenciador de domínio quando você define a estação de trabalho no banco de dados. Essa estação de trabalho é registrada no banco de dados Tivoli Workload Scheduler como **gerenciador**.

Gerenciador de domínio dinâmico
Um componente instalado em uma rede do Tivoli Workload Scheduler distribuída que é o hub de gerenciamento em um domínio. Toda a comunicação para e a partir de agentes no domínio é roteada por meio do gerenciador de domínio dinâmico. Ao instalar um gerenciador de domínio dinâmico os seguintes tipos de estação de trabalho são criados no banco de dados:

fta Componente de agente tolerante a falhas configurado manualmente como gerenciador de domínio

intermediário
Componente do servidor de broker

agent Componente de agente dinâmico

Backup gerenciador de domínio dinâmico
Uma estação de trabalho pode atuar como um backup para o gerenciador de domínio dinâmico quando ocorrerem problemas. Ela é efetivamente um gerenciador de domínio dinâmico, aguardando para ser ativado. Seu uso é opcional. Saiba mais sobre a comutação para um gerenciador de domínio dinâmico de backup no *Tivoli Workload Scheduler Administration Guide*. Ao instalar um gerenciador de domínio dinâmico os seguintes tipos de estação de trabalho são criados no banco de dados:

fta Componente de agente tolerante a falhas.

intermediário
Componente do servidor de broker

agent Componente de agente dinâmico

Agente tolerante a falhas
Uma estação de trabalho que recebe e executa tarefas. Se houver algum problema de comunicação com seu gerenciador de domínio, ela pode executar tarefas localmente. Ela é instalada como um agente e, em seguida, configurada como uma estação de trabalho do agente tolerante a falhas quando você define a estação de trabalho no banco de dados. Essa estação de trabalho é registrada no banco de dados Tivoli Workload Scheduler como **fta**.

Agente padrão
Uma estação de trabalho que recebe e executa tarefas apenas sob o controle de seu gerenciador de domínio. Ela é instalada como um agente e, em seguida, configurada como uma estação de trabalho do agente padrão quando você define a estação de trabalho no banco de dados. Essa estação de trabalho é registrada no banco de dados Tivoli Workload Scheduler como **s-agent**.

Agente Estendido
Uma estação de trabalho na qual um método de acesso do Tivoli Workload Scheduler for Applications foi instalado como uma ponte para que você possa planejar tarefas no SAP R/3, Oracle E-Business Suite, PeopleSoft, z/OS ou aplicativos customizados. Ela deve ser fisicamente hospedada por um gerenciador de domínio principal, um gerenciador de domínio, um agente padrão ou um agente tolerante a falhas (até 255 agentes estendidos por agente tolerante a falhas) e, em seguida, definida como um agente estendido no banco de dados. Para obter informações adicionais, consulte o Guia do Usuário do *Tivoli Workload Scheduler for Applications*. Essa estação de trabalho é registrada no banco de dados Tivoli Workload Scheduler como **x-agent**.

Intermediário da Carga de Trabalho
Uma estação de trabalho que executa tipos de tarefas existentes e tipos de tarefas com opções avançadas. É o servidor do broker instalado com o gerenciador de domínio principal e ogerenciador de domínio dinâmico. É possível hospedar uma ou mais das estações de trabalho a seguir:

- Agente Estendido
- Mecanismo remoto
- Conjunto

- Conjunto dinâmico
- Agente. Essa definição inclui os seguintes agentes:
  - Agente
  - Agente do Tivoli Workload Scheduler for z/OS
  - Agente para z/OS

  Para obter informações adicionais sobre oagent e oAgente do Tivoli Workload Scheduler for z/OS, consulte *Planejando Carga de Trabalho Dinamicamente*. Para obter informações adicionais sobre oAgent for z/OS, consulte *Planejando com o Agent for z/OS*.

Essa estação de trabalho é registrada no banco de dados Tivoli Workload Scheduler como **broker**.

Agente dinâmico
U ma estação de trabalho que gerencia vários tipos de tarefas, por exemplo, tarefas de banco de dados ou de FTP específicas, além dos tipos de tarefas existentes. Esta estação de trabalho é automaticamente criada e registrada no banco de dados do Tivoli Workload Scheduler durante a instalação do agent. O agent é hospedado pela workload broker workstation. Como os processos de instalação e registro são executados automaticamente, ao visualizar oagent no Dynamic Workload Console, ele resulta como atualizado no Resource Advisor Agent. É possível agrupar agentes em conjuntos e conjuntos dinâmicos. Essa estação de trabalho é registrada no banco de dados Tivoli Workload Scheduler como **agente**.

Em uma configuração simples, os agentes dinâmicos são conectados diretamente a um gerenciador de domínio principal ou a um gerenciador de domínio dinâmico. No entanto, em topologias de rede mais complexas, se a configuração da rede impedir o gerenciador de domínio principal ou o gerenciador de domínio dinâmico de se comunicar diretamente com o agente dinâmico, é possível configurar seus agentes dinâmicos para utilizarem um gateway local ou remoto.

Nota: Se você tiver a opção global `enAddWorkstation` configurada como `"yes"`, a definição de estação de trabalho do agente dinâmico é automaticamente incluído no Plano após o processo de instalação criar a estação de trabalho do agente dinâmico no banco de dados.

Conjunto
Uma estação de trabalho lógica que agrupa um conjunto de agentes com características de hardware ou software similares para o qual as tarefas são enviadas. O Tivoli Workload Scheduler balanceia as tarefas entre os agentes do conjunto e redesigna tarefas automaticamente a agentes disponíveis se um agent não estiver mais disponível. Para criar um conjunto de agentes em seu ambiente do Tivoli Workload Scheduler, defina uma estação de trabalho do tipo**pool** hospedada pela workload broker workstation, em seguida, selecione os agentes que deseja incluir no conjunto. É possível definir o conjunto usando oDynamic Workload Console ou o comando **composer**. Essa estação de trabalho é registrada no banco de dados Tivoli Workload Scheduler como **conjunto**. Ao criar um conjunto no ambiente do Tivoli Workload Scheduler, um recurso lógico com o mesmo nome é criado automaticamente no Dynamic Workload Broker. Este recurso lógico é usado para correlacionar e agrupar os agentes pertencentes ao mesmo conjunto, e como um requisito para as tarefas planejadas no conjunto do Tivoli Workload Scheduler. Considere que esses objetos de banco de dados

são dois objetos diferentes. Se você renomear o conjunto do Tivoli Workload Scheduler, esta mudança não será feita no recurso lógico do Dynamic Workload Broker.

Conjunto dinâmico
Uma estação de trabalho lógica que agrupa um conjunto de agentes, que é dinamicamente definida com base nos requisitos de recurso especificados e especificada pela workload broker workstation. Por exemplo, se precisar de uma estação de trabalho com um uso de CPU baixo e o Windows instalado para executar sua tarefa, você especifica esses requisitos usando oDynamic Workload Console ou o comando **composer**. Ao salvar o conjunto de requisitos, uma nova estação de trabalho é criada automaticamente no banco de dados do Tivoli Workload Scheduler. Esta estação de trabalho mapeia todo oagentes em seu ambiente que atende aos requisitos especificados. O conjunto resultante é atualizado dinamicamente sempre que um novoagent adequado for disponibilizado. Tarefas planejadas nessa estação de trabalho herdam automaticamente os requisitos definidos para a estação de trabalho. Essa estação de trabalho é hospedada pela workload broker workstation e registrada no banco de dados do Tivoli Workload Scheduler como **d-pool**.

Mecanismo remoto
Uma estação de trabalho que gerencia a troca de informações sobre resolução de dependências cruzadas entre seu ambiente e um mecanismo remoto do Tivoli Workload Scheduler for z/OS (controlador) ou mecanismo do Tivoli Workload Scheduler (gerenciador de domínio principal ou gerenciador do domínio backup principal). Essa estação de trabalho é hospedada pela workload broker workstation e registrada no banco de dados do Tivoli Workload Scheduler como **rem-eng**.

Nota: Se planeja alterar os tipos de estação de trabalho, considere as regras a seguir:

- É possível alterar o agente tolerante a falhas, o agente padrão, o agente estendido, o gerenciador de domínio e as estações de trabalho do broker de carga de trabalho dinâmico para qualquer tipo de estação de trabalho, com exceção do agente dinâmico, conjunto, conjunto dinâmico e mecanismo remoto.
- Não é possível alterar o tipo de agente dinâmico, conjunto, conjunto dinâmico e mecanismo remoto.

Para obter informações sobre como definir estações de trabalho, consulte "Definição de estação de trabalho" na página 139.

Classe de Estação de Trabalho

Estações de trabalho podem ser agrupadas em classes. Uma *classe de estação* é um grupo de estações de trabalho com características de planejamento de tarefas semelhantes. Qualquer número de estações de trabalho pode ser agrupado em uma classe e uma estação de trabalho pode estar em várias classes. Tarefas e fluxos de tarefas podem ser designados para execução em uma classe de estação de trabalho específica, e isso facilita a execução de tarefas e de fluxos de tarefas em diversas estações de trabalho.

Por exemplo, é possível configurar os seguintes tipos de classes de estação de trabalho:

- Classes de estação de trabalho que agrupam estações de trabalho de acordo com sua estrutura departamental interna, para que você possa definir uma tarefa para execução em todas as estações de trabalho em um departamento

- Classes de estação de trabalho que agrupam estações de trabalho de acordo com o software instalado nelas, para que você possa definir uma tarefa para execução em todas as estações de trabalho que possuem um aplicativo específico instalado
- Classes de estação de trabalho que agrupam estações de trabalho de acordo com a função do usuário, para que você possa definir uma tarefa para execução em todas as estações de trabalho pertencentes a, por exemplo, gerenciadores

Neste exemplo, uma estação de trabalho individual pode estar em uma classe de estação de trabalho para seu departamento, outra para seu usuário e várias para o software instalado nela.

Estações de trabalho também podem ser agrupadas em domínios. Isto é feito quando sua rede é configurada. O nome do domínio não é um dos critérios de seleção quando se escolhe onde executar uma tarefa, portanto, talvez você precise espelhar a estrutura do domínio com classes da estação de trabalho se quiser planejar uma tarefa para execução em todas as estações de trabalho em um domínio.

Para obter informações adicionais sobre domínios, consulte "Domínio"

Para obter mais informações sobre como definir classes da estação de trabalho, consulte "Definição de classe de estação de trabalho" na página 153.

Domínio

Todas as estações de trabalho em uma rede do Tivoli Workload Scheduler distribuída são organizadas em um ou mais *domínios*, sendo que cada um consiste em um ou mais agentes e um gerenciador de domínio que agem como o hub de gerenciamento. A maior parte da comunicação para o agente e a partir dele no domínio é encaminhada por meio do gerenciador de domínio.

Todas as redes têm um domínio principal onde o gerenciador de domínio é o gerenciador de domínio principal. Ele mantém o banco de dados de todos os objetos de planejamento no domínio e nos arquivos de configuração centrais. O gerenciador de domínio principal gera o plano e cria e distribui o arquivo Symphony. Além disso, os logs e relatórios da rede são mantidos no gerenciador de domínio principal.

É possível organizar todos os agentes em sua rede em um único domínio ou em vários domínios.

Redes de domínio único

Uma rede de domínio único consiste em um gerenciador de domínio principal e em qualquer número de agentes. A seguir temos um exemplo de uma rede de domínio único. Uma rede de domínio único é adequada para empresas que têm poucos locais e funções negócios. Todas as comunicações na rede são roteadas através do gerenciador de domínio principal. Como um único local, você se preocupa apenas com a confiabilidade da rede local e com a quantidade de tráfego que ela pode manipular.

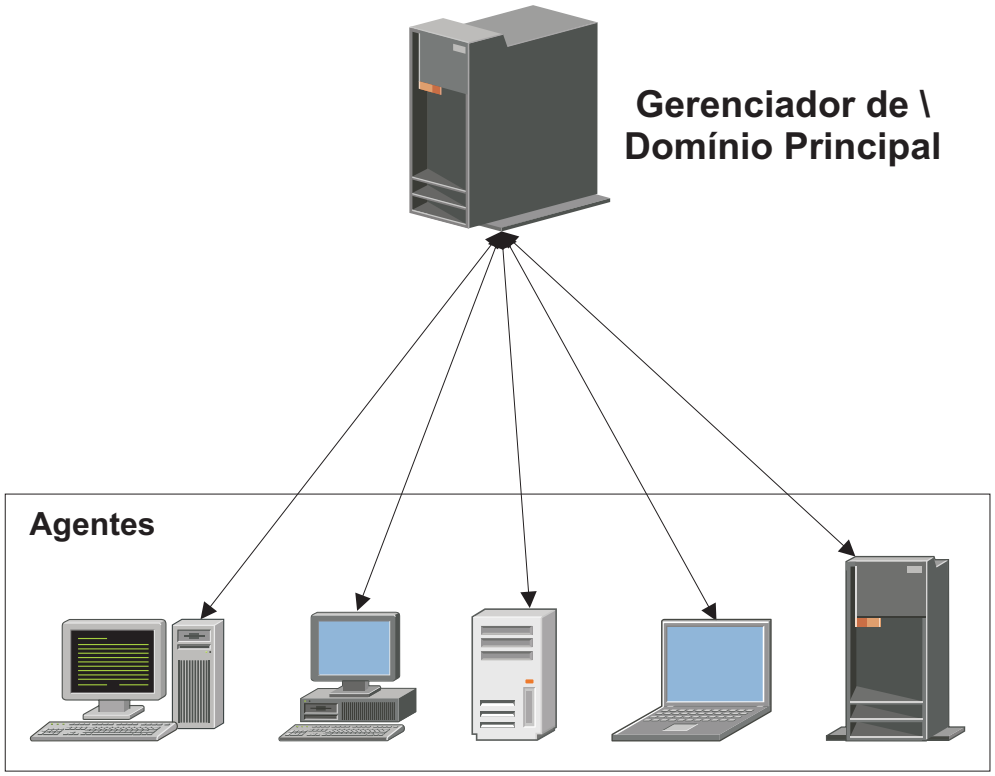


Figura 1. Rede de Domínio Único

Rede de domínios múltiplos

Redes de domínios múltiplos são adequadas especialmente para empresas que abrangem vários locais, departamentos ou funções de negócios. Uma rede de domínios múltiplos consiste em um gerenciador de domínio principal, em qualquer número de gerenciadores de domínio de camada inferior e qualquer número de agentes em cada domínio. Agentes se comunicam apenas com gerenciadores de domínio, e os gerenciadores de domínio se comunicam com seus gerenciadores de domínio pais. A hierarquia de domínios pode ir abaixo de qualquer número de níveis.

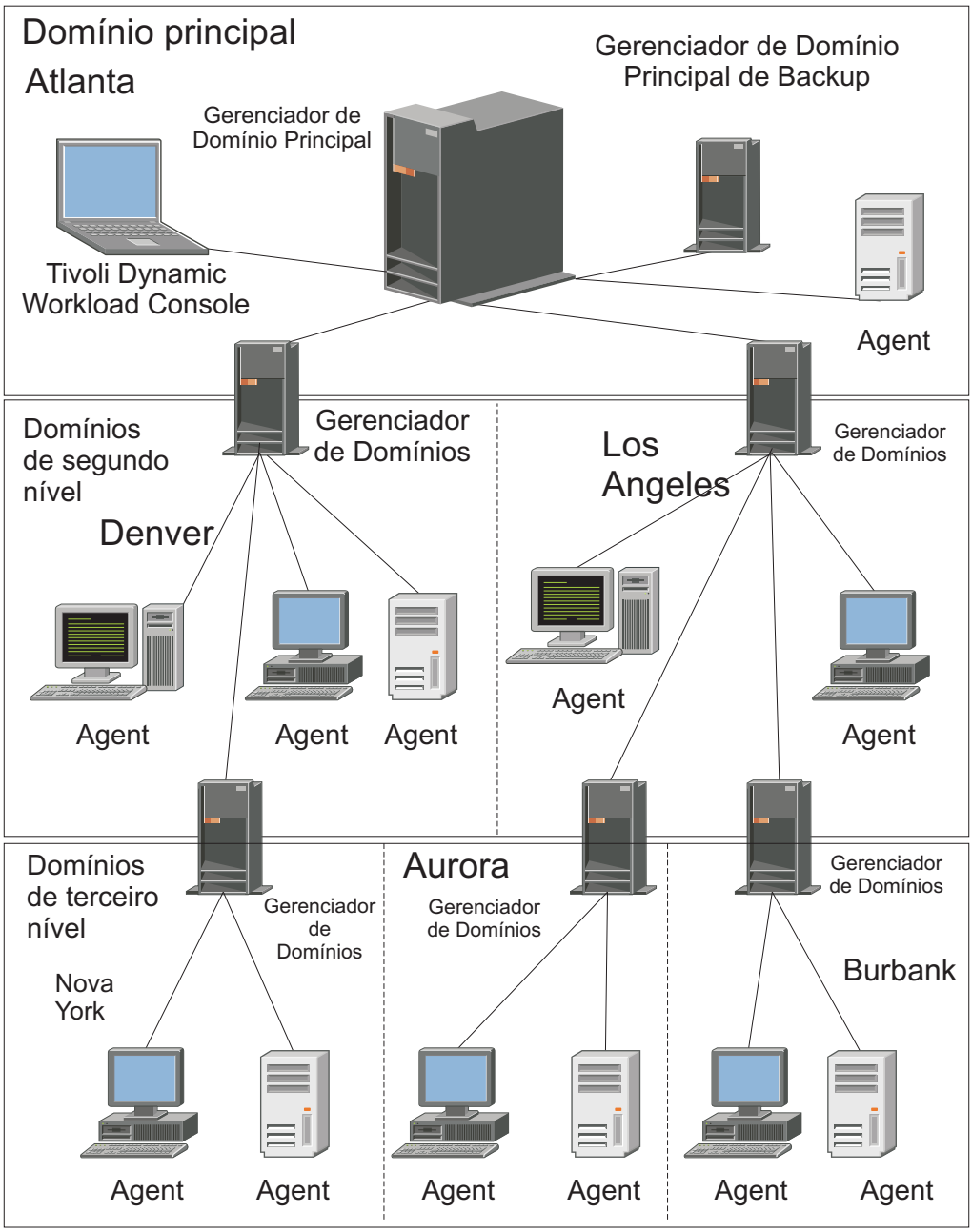


Figura 2. Rede de domínios múltiplos

Neste exemplo, o gerenciador de domínio principal está localizado em Atlanta. O gerenciador de domínio principal contém os arquivos de banco de dados utilizados para documentar os objetos de planejamento e distribui o arquivo Symphony para seus agentes e gerenciadores de domínio principal em Denver e Los Angeles. Os gerenciadores de domínio de Denver e Los Angeles distribuem então o arquivo Symphony para seus agentes e gerenciadores de domínio subordinados em Nova Iorque, Aurora e Burbank. O gerenciador de domínio principal em Atlanta é responsável pela difusão de informações entre domínios por toda a rede.

Todas as comunicações para e do gerenciador de domínio Boulder são roteadas através de seu gerenciador de domínio pai em Denver. Se houver algum planejamento ou tarefa no domínio Boulder dependente de

planejamentos ou tarefas no domínio Aurora, essas dependências serão resolvidas pelo gerenciador de domínio de Denver. A maioria das dependências interagentes é manipulada localmente pelos gerenciadores de domínio de camada inferior, reduzindo significativamente o tráfego na rede.

É possível alterar a infra-estrutura do domínio dinamicamente, conforme desenvolve sua rede. Para mover uma estação de trabalho para um domínio diferente, você precisa alterar o nome do domínio em sua definição de banco de dados.

Tabela 5. Dica

Dica:	Não é possível planejar tarefas ou fluxos de tarefas em todas as estações de trabalho em um domínio, identificando o domínio na definição da tarefa ou do fluxo de tarefas. Para conseguir isso, crie uma *classe de estação de trabalho* que contenha todas as estações de trabalho no domínio.

Para obter mais informações sobre classes de estação de trabalho, consulte "Classe de Estação de Trabalho" na página 14.

Para obter informações sobre como definir domínios, consulte "Definição de domínio" na página 154.

Regra de Evento

Uma *regra de evento* define um conjunto de ações que são executadas quando ocorrem condições de evento específicas. Uma definição de regra de evento correlaciona eventos e ações do acionador.

Para obter informações sobre como definir regras de eventos, consulte "Definindo Regras de Eventos" na página 126.

Resource

Um *recurso* é um recurso do sistema lógico ou físico que você utiliza como uma dependência para tarefas e fluxos de tarefas. Uma tarefa ou um fluxo de tarefas com uma dependência de recurso não podem ser iniciados até que a quantidade necessária do recurso definido esteja disponível.

Para obter informações sobre como definir recursos, consulte "Definição do recurso" na página 206.

Parâmetro

Um *parâmetro* é um objeto ao qual você designa valores diferentes para serem substituídos em tarefas e fluxos de tarefas, de valores no banco de dados ou no tempo de execução. Parâmetros são úteis quando você tem valores que mudam dependendo da tarefa ou do fluxo de tarefas. Definições de tarefa e fluxo de tarefas que utilizam parâmetros são atualizadas automaticamente com o valor no início do de produção. Utilize parâmetros como substitutos para valores repetitivos ao definir tarefas e fluxos de tarefas. Por exemplo, o uso de parâmetros para logon de usuário e nome de arquivo de script em definições de tarefa e para dependências de arquivo e prompt permite o uso de valores que podem ser mantidos centralmente no banco de dados no principal.

Para obter mais informações sobre como definir parâmetros, consulte "Definição de Variável e Parâmetro" na página 198.

Usuário

Um *Usuário* é o nome de usuário usado como o valor de logon para várias definições de tarefa do sistema operacional. Os usuários devem estar definidos no banco de dados.

Se você planejar uma tarefa em um agente, em um conjunto ou em um conjunto dinâmico, a tarefa será executada com o usuário definido no conjunto ou conjunto dinâmico. Todavia, o usuário deve existir em todas as estações de trabalho no conjunto ou conjunto dinâmico em que planeja executar a tarefa.

Nota: Se você tiver a opção global `enAddUser` configurada como `"yes"`, a definição de usuário será automaticamente incluída no plano após você criar ou modificar a definição de usuário no banco de dados.

Tabela de Variáveis

Uma *tabela de variáveis* é uma tabela que contém diversas variáveis e seus valores. Todos os parâmetros globais, agora chamados de *variáveis*, estão contidos em pelo menos uma tabela de variáveis.

Não é necessário criar tabelas de variáveis para poder utilizar variáveis, pois o planejador fornece uma tabela de variáveis padrão.

Entretanto, talvez você queira definir uma variável com o mesmo nome, mas valores diferentes, dependendo de quando e onde ela será utilizada. Isso é feito designando-se valores diferentes para a mesma variável em tabelas de variáveis diferentes. É possível então utilizar o mesmo nome de variável em definições de tarefa diferentes e ao definir dependências de arquivos e avisos. Tabelas de variáveis podem ser designadas no ciclo de execução, fluxo de tarefas e nível da estação de trabalho.

As tabelas de variáveis podem ser particularmente úteis em definições de tarefas quando uma definição de tarefa é utilizada como um modelo para uma tarefa que pertence a mais de um fluxo de tarefas. Por exemplo, é possível designar valores diferentes para a mesma variável e reutilizar a mesma definição de tarefa em fluxos de tarefas diferentes.

Para obter informações sobre como definir tabelas de variáveis, consulte “Definição da Tabela de Variáveis” na página 202.

A Rede do Tivoli Workload Scheduler

Uma rede do Tivoli Workload Scheduler consiste em um conjunto de ***estações de trabalho*** vinculadas no qual você executa processamento de tarefa do batch utilizando as capacidades de gerenciamento do Tivoli Workload Scheduler.

As estações de trabalho comunicam-se utilizando links de TCP/IP e uma tecnologia de armazenamento e redirecionamento para manter a consistência e a tolerância a falhas em toda a rede. Isso significa que se uma estação de trabalho não estiver vinculada, todas as informações são armazenadas no arquivo de mensagens e enviadas somente quando o link for restabelecido.

A rede do Tivoli Workload Scheduler consiste em um ou mais domínios, cada um tendo uma estação de trabalho de ***gerenciador de domínio***, que age como um hub de gerenciamento, e uma ou mais estações de trabalho do ***agente***.

Existem quatro tipos de agente: ***padrão***, ***tolerante a falhas***, ***estendido*** e ***do workload broker***. Agentes padrão e tolerantes a falhas podem ser definidos nos computadores UNIX e Windows. Os agentes estendidos são definições lógicas, cada um hospedado por uma estação de trabalho física, e são utilizados para executar o processamento de tarefa quando um agente não está instalado. Por exemplo, agentes estendidos estão disponíveis para Peoplesoft, SAP R/3, z/OS, CA-7, JES, OPC, Oracle EBS e VMS, mas também é possível instalá-los em sistemas UNIX e Windows. Os agentes do Workload Broker são estações de trabalho que gerenciam o ciclo de vida das tarefas de tipo do Tivoli Workload Scheduler Workload Broker no Tivoli dynamic workload broker.

Outro tipo de estação de trabalho que é possível definir em sua rede é uma ***estação de trabalho de mecanismo remoto***. Esse tipo e estação de trabalho é usado para gerenciar a comunicação com um mecanismo remoto do Tivoli Workload Scheduler, distribuído ou baseado no z/OS, para gerenciar dependências para tarefas locais de tarefas definidas no mecanismo remoto. Para obter informações adicionais, consulte a Capítulo 19, “Definindo e Gerenciando Dependências Cruzadas”, na página 625.

Para obter informações sobre estações de trabalho, consulte “Definição de estação de trabalho” na página 139.

Na topologia hierárquica do Tivoli Workload Scheduler, o ***gerenciador de domínio principal*** é o gerenciador de domínios do domínio mais acima. Todas as tarefas de configuração da produção e a geração do ***plano de produção*** são executadas no gerenciador de domínio principal. Um plano de produção contém todas as atividades de gerenciamento de tarefa a serem executadas na rede do Tivoli Workload Scheduler durante um espaço de tempo específico. Uma cópia do plano de produção é distribuída do gerenciador de domínio principal para as outras estações de trabalho. Em cada estação de trabalho, o Tivoli Workload Scheduler ativa e monitora suas próprias tarefas e envia o status de processamento de tarefa para o gerenciador de domínio principal.

Para obter informações adicionais sobre as capacidades de gerenciamento de plano do Tivoli Workload Scheduler, consulte o Capítulo 4, “Gerenciando o Ciclo de Produção”, na página 57.

Configurando seu Ambiente de Tempo de Execução do Tivoli Workload Scheduler

Esta seção fornece uma visão geral de alto nível de como é possível configurar seu ambiente de tempo de execução do Tivoli Workload Scheduler.

Configurando Propriedades

É possível configurar dois tipos de propriedades para configurar seu ambiente de tempo de execução do Tivoli Workload Scheduler, propriedades configuradas no gerenciador de domínio principal e que afetam o processamento em todas as estações de trabalho na rede do Tivoli Workload Scheduler e as propriedades que são configuradas localmente em uma estação de trabalho e que afetam o processamento somente nessa estação de trabalho. As primeiras são gerenciadas utilizando o programa de linha de comandos do Tivoli Workload Scheduler denominado **optman** e as demais são definidas localmente na estação de trabalho customizando os arquivos de configuração **useropts**, **localopts** e **jobmanrc**.

Para obter informações adicionais sobre como utilizar a linha de comandos **optman** para gerenciar opções globais e sobre opções locais definidas no arquivo `localopts`, consulte o *Guia de Administração do IBM Tivoli Workload Scheduler*.

Para obter informações adicionais sobre as opções locais definidas no arquivo `useropts`, consulte "Configurando Opções para Utilização das Interfaces com o Usuário" na página 53.

Configurando a Segurança

Toda vez que você executar um programa do Tivoli Workload Scheduler ou chamar um comando do Tivoli Workload Scheduler, as informações de segurança são lidas a partir de um arquivo especial, o ***arquivo de Segurança***, para determinar suas capacidades de usuário. Esse arquivo contém uma ou mais *definições de usuário*. Uma definição de usuário é um grupo de um ou mais usuários que são permitidos ou rejeitados para desempenhar ações específicas nos tipos de objeto de planejamento específicos.

O usuário principal do Tivoli Workload Scheduler, ***TWS_user***, é definido no tempo de instalação no arquivo de segurança. Esse ID do usuário pode ser utilizado para concluir o procedimento de configuração, para configurar propriedades e para gerenciar definições de usuários no arquivo de segurança. É possível modificar o arquivo de segurança a qualquer momento para atender os requisitos do sistema.

Para obter informações adicionais sobre como gerenciar autorizações do usuário, consulte *Guia de Administração do Tivoli Workload Scheduler*.

Definindo Atividades de Planejamento Usando o Tivoli Workload Scheduler

Para executar atividades de planejamento utilizando o Tivoli Workload Scheduler, é necessário primeiro definir o ambiente que você deseja gerenciar em termos de *objetos de planejamento* e em termos de regras a serem aplicadas ao planejar operações para executar nesses objetos. Estas informações são armazenadas pelo Tivoli Workload Scheduler em um Banco de Dados DB2 ou Oracle Relacional, doravante chamado de ***banco de dados***.

Além das definições dos objetos de planejamento, como as tarefas, os fluxos de tarefas, os recursos, as estações de trabalho e assim por diante, o banco de dados também contém estatísticas sobre tarefas processadas e fluxos de tarefas, assim como informações sobre o usuário que criou um objeto e quando um objeto foi modificado pela última vez. É possível gerenciar o planejamento de definições de objetos no banco de dados usando o programa da linha de comandos do Tivoli Workload Scheduler chamado **composer** ou as interface gráfica com o usuário, o **Dynamic Workload Console**. É possível recuperar estatísticas ou informações de histórico sobre tarefas e fluxos de tarefas processados no banco de dados, utilizando:

- Os **utilitários de relatório** do Tivoli Workload Scheduler da linha de comandos.
- O Dynamic Workload Console.
- As visualizações de banco de dados.

Para obter mais informações sobre como definir objetos de planejamento, consulte Capítulo 8, "Definindo Objetos no Banco de Dados", na página 137.

Para obter informações adicionais sobre os comandos dos utilitários, consulte o Capítulo 15, "Obtendo Relatórios e Estatísticas", na página 559.

Para obter informações adicionais sobre a Dynamic Workload Console, consulte a documentação correspondente.

Para obter informações adicionais sobre as visualizações de banco de dados, consulte o *IBM Tivoli Workload Scheduler: Visualizações de Banco de Dados*.

Controlando a Tarefa e o Processamento do Fluxo de Tarefas

É possível controlar como as tarefas e os fluxos de tarefas são processados configurando uma ou mais das seguintes regras:

Definindo Dependências

Uma ***dependência*** é um pré-requisito que deve ser satisfeito para que o processamento possa continuar. É possível definir dependências para tarefas e fluxos de tarefas para assegurar a ordem correta de processamento. Em seu ambiente de planejamento distribuído do Tivoli Workload Scheduler, é possível escolher entre quatro tipos de dependências:

- ***Na conclusão de tarefas e fluxos de tarefas***: uma tarefa ou um fluxo de tarefas, denominado *successor*, não deve iniciar o processamento até outras tarefas e fluxos de tarefas, denominados *predecessor*, terem sido concluídos com sucesso. Para obter informações adicionais, consulte a "follows" na página 232.
- ***Recurso***: uma tarefa ou um fluxo de tarefas precisa de um ou mais recursos disponíveis antes de poder começar a execução. Para obter informações adicionais, consulte o "needs" na página 243.
- ***Arquivo***: uma tarefa ou um fluxo de tarefas precisa ter acesso a um ou mais arquivos antes de poder iniciar a execução. Para obter informações adicionais, consulte o "opens" na página 249.
- ***Prompt***: uma tarefa ou um fluxo de tarefas precisa aguardar uma resposta afirmativa a um prompt antes de poder iniciar o processamento. Para obter informações adicionais, consulte "Definição do Prompt" na página 204 e "prompt" na página 252.

É possível definir até 40 dependências para uma tarefa ou um fluxo de tarefas.

Em uma rede do Tivoli Workload Scheduler, as dependências podem cruzar limites da estação de trabalho. Por exemplo, é possível tornar `job1`, que é executado no ambiente local do Tivoli Workload Scheduler `site1`, dependente da conclusão bem-sucedida de `job2`, que é executado em um ambiente remoto do Tivoli Workload Scheduler `site2`. O ambiente de planejamento remoto pode ser mecanismos do Tivoli Workload Scheduler for z/OS (controlador) ou outros mecanismos do Tivoli Workload Scheduler (gerenciador de domínio principal). Dois tipos de dependências implementam tal requisito:

Dependência de Interligação de Redes

É uma implementação simples e distribuída. Use esse tipo de dependência quando:

- O ambiente local do Tivoli Workload Scheduler for distribuído.
- Você quiser procurar uma instância da tarefa predecessora remota somente no plano atualmente em execução (plano de produção) no ambiente remoto.
- For necessário corresponder uma instância predecessora do plano remoto, não *essa* instância predecessora específica.
- For possível esperar o intervalo de pesquisa expirar antes de ser atualizado sobre a transição do status da tarefa remota.

- Você não se importar de usar diferentes sintaxes e configurações com base em se o ambiente remoto do Tivoli Workload Scheduler é distribuído, em vez de z/OS.
- Você não se importar de usar um protocolo de conexão proprietário para se comunicar com o mecanismo remoto.

Para obter informações adicionais, consulte a Capítulo 18, "Gerenciando Dependências da Interligação de Redes", na página 615.

Dependência cruzada
É uma implementação mais abrangente e completa. Use esse tipo de dependência quando:

- Seu ambiente local do Tivoli Workload Scheduler puder ser distribuído ou z/OS.
- Você quiser procurar a instância predecessora remota também entre as instâncias planejadas que ainda não estão incluídas no plano atualmente em execução no mecanismo remoto.
- Você quiser corresponder a instância predecessora no plano do mecanismo remoto. Para fazer isso, é possível usar diferentes critérios de correspondência prontos para uso.
- Você deseja que sua dependência seja atualizada assim que a instância da tarefa remota mude de status. Para fazer isso, o produto usa uma notificação assíncrona do mecanismo remoto para o mecanismo local.
- Você deseja usar a mesma sintaxe e configuração, independentemente de se o ambiente local do Tivoli Workload Scheduler é distribuído ou z/OS.
- Você deseja usar conexões HTTP ou HTTPS para comunicação com o mecanismo remoto.

Para obter informações adicionais, consulte a Capítulo 19, "Definindo e Gerenciando Dependências Cruzadas", na página 625.

Configurando Restrições de Tempo

As ***restrições de tempo*** podem ser especificadas para tarefas e fluxos de tarefas. Para um ciclo de execução específico, é possível especificar a hora na qual o processamento é iniciado, utilizando a palavra-chave **at**, ou a hora após a qual o processamento não é mais iniciado, utilizando a palavra-chave **until**. Especificando ambas, você define uma janela de tempo dentro da qual uma tarefa ou um fluxo de tarefa é executado. Ambas, **at** e **until**, representam dependências de hora.

Outra configuração de hora que pode ser especificada é **schedtime**; indica a hora referida ao calcular dependências de tarefas e de fluxos de tarefas. Também é possível especificar uma ***taxa de repetição***; por exemplo, é possível fazer com que o Tivoli Workload Scheduler ative a mesma tarefa a cada 30 minutos entre às 8h30 e às 13h30.

Também é possível especificar uma **duração máxima** ou uma **duração mínima** para uma tarefa definida em um fluxo de tarefas. Se uma tarefa estiver em execução e o tempo de duração máxima tiver sido excedido, a tarefa poderá ser eliminada ou poderá continuar em execução. Se uma tarefa não for executada o suficiente para atingir o tempo de duração mínima especificado, a tarefa poderá ser configurada como status de Encerramento Anormal, como status Confirmar aguardando confirmação do usuário ou pode continuar em execução.

Nota: Especificar as durações máxima e mínima para uma tarefa definida em um fluxo de tarefas resulta em um arquivo Symphony que ocupa 512 bytes a mais do que uma tarefa sem estas especificações.

Para obter informações adicionais, consulte “às” na página 220, “prazo final” na página 223, “a cada” na página 225, “schedtime” na página 253, “until” na página 256, “maxdur” na página 240 e “mindur” na página 241.

Configurando Prioridade da Tarefa e Limitação da Estação de Trabalho

O Tivoli Workload Scheduler tem seu próprio sistema de enfileiramento, que consiste em níveis de ***prioridade***. Designar uma prioridade às tarefas e aos fluxos de tarefas fornece controle adicional sobre suas precedências e a ordem de execução.

A ***proteção de tarefa*** fornece outro tipo de controle sobre o processamento de tarefa em uma estação de trabalho. Quando configurado para um nível de prioridade, ele permite que apenas tarefas e fluxos de tarefas cuja prioridade exceda o fence da tarefa sejam executados nessa estação de trabalho. Definir o limitador como 40, por exemplo, evita que os jobs com prioridades de 40 ou menos sejam ativados.

Para obter informações adicionais, consulte “fence” na página 380 e “prioridade” na página 251.

Configurando Limites

O ***limite*** fornece uma maneira de configurar o número mais alto de tarefas que o Tivoli Workload Scheduler tem permissão para executar. Você pode configurar um limite:

- Na definição do fluxo de tarefas que utiliza o argumento ***limite de tarefa***
- Na definição da estação de trabalho, utilizando o comando ***limit cpu***

Configurar o limite em uma estação de trabalho para 25, por exemplo, permite que o Tivoli Workload Scheduler não tenha mais de 25 tarefas em execução simultaneamente nessa estação de trabalho.

Para obter informações adicionais, consulte “limit cpu” na página 384 e “limit sched” na página 385.

Definindo Recursos

É possível definir ***recursos*** para apresentar recursos físicos ou lógicos em seu sistema. Cada recurso é representado por um nome e um número de unidades disponíveis. Se você tiver três unidades de fita, por exemplo, poderá definir um recurso denominado `tapes` com três unidades disponíveis. Uma tarefa que utiliza duas unidades do recurso `tapes` impediria, então, que outras tarefas, que requerem mais do que a unidade restante, fossem ativadas. No entanto, como um recurso não é vinculado estritamente a um recurso, é possível utilizar um recurso fictício como uma dependência para controlar o processamento de tarefas.

Para obter informações adicionais, consulte o “Definição do recurso” na página 206.

Solicitando Confirmação de Tarefa

Pode haver cenários em que o status de conclusão de uma tarefa não possa ser determinado até que você tenha executado algumas tarefas. Você talvez queira verificar os resultados impressos em um relatório, por exemplo. Nesse caso, é possível configurar na definição de tarefa que a tarefa requer *confirmação* e o Tivoli Workload Scheduler aguarda sua resposta antes de marcar a tarefa como bem-sucedida ou em falha.

Para obter informações adicionais, consulte o “confirm” na página 372.

Definindo Ações de Recuperação da Tarefa

Ao planejar uma tarefa, é possível especificar o tipo de recuperação que você deseja executar pelo Tivoli Workload Scheduler se a tarefa falhar. As opções de recuperação predefinidas são:

- Continuar com a próxima tarefa.
- Parar e não iniciar o próximo.
- Executar a tarefa que falhou novamente.

Além disso, é possível especificar outras ações a serem tomadas em termos de tarefas de recuperação e prompts de recuperação. Por exemplo, se uma tarefa falhar, é possível que o Tivoli Workload Scheduler execute automaticamente uma tarefa de recuperação, emita um prompt de recuperação que requer uma resposta afirmativa e, em seguida, execute a tarefa em falha novamente.

Para obter informações adicionais, consulte o “Tarefa” na página 711.

Gerenciando Atividades de Planejamento de Produção com o Tivoli Workload Scheduler

Toda vez que um novo plano de produção é gerado, o Tivoli Workload Scheduler seleciona os fluxos de tarefas que são executados na janela de tempo especificada para o plano e transporta o saldo de fluxos de tarefas incompletos do plano de produção anterior. Todas as informações necessárias são gravadas em um arquivo, chamado ***Symphony***, que é continuamente atualizado durante o processamento para indicar trabalho concluído, trabalho em andamento e trabalho a ser feito. O programa da linha de comandos do Tivoli Workload Scheduler **conman** (Console Manager) é usado para gerenciar as informações no arquivo `Symphony`. O programa de linha de comandos **conman** pode ser utilizado para:

- Iniciar e parar processos de controle do Tivoli Workload Scheduler.
- Exibir o status das tarefas e dos fluxos de tarefas.
- Alterar prioridades e dependências.
- Alterar a proteção de tarefa e os limites de tarefa.
- Executar as tarefas novamente.
- Cancelar tarefas e fluxos de tarefa.
- Enviar novas tarefas e fluxos de tarefa.
- Responder a prompts.
- Vincular e desvincular estações de tarefa na rede do Tivoli Workload Scheduler.
- Modificar o número de recursos disponíveis.

A partir da versão 9.1, todas as informações de plano gravadas no arquivo `Symphony` são então replicadas no banco de dados. Várias operações de monitoramento solicitadas do Dynamic Workload Console acessam o banco de dados, em vez do arquivo Symphony, resultando em tempos de resposta e desempenho geral mais rápidos. As seguintes operações solicitadas do Dynamic Workload Console acessam informações do banco de dados:

- Monitorando tarefas e fluxos de tarefas
- Atualizando visualizações de monitoramento de tarefas e de fluxos de tarefas
- Monitorando estações de trabalho
- Monitorando recursos, arquivos, prompts
- Executando relatórios de linha de base

- Exibindo o plano na visualização gráfica
- Exibindo uma visualização de impacto

Automatizando a Carga de Trabalho Usando Regras de Eventos

Além de fazer o planejamento de tarefas baseado em planos, é possível automatizar a carga de trabalho com base na demanda, com o auxílio de regras de eventos. O objetivo das regras de eventos é executar um conjunto predefinido de ações em resposta a eventos específicos que afetam objetos Tivoli Workload Scheduler e não Tivoli Workload Scheduler.

Com relação aos objetos Tivoli Workload Scheduler, o produto fornece um plug-in que pode ser utilizado para detectar os seguintes eventos:

- Uma tarefa ou fluxo de tarefas específicos:
  - Altera o status
  - Está além da hora de início mais recente
  - Foi enviado
  - Foi cancelado
  - Foi reiniciado
  - Torna-se atrasado
- Uma determinada estação de trabalho:
  - Altera o status
  - Altera seu status de link a partir da estação de trabalho pai
  - Altera seu status de link a partir da estação de trabalho filha
- Um aviso específico é exibido ou respondido
- O servidor de aplicativos em uma determinada estação de trabalho foi iniciado ou interrompido.

Quando ocorrer algum desses eventos, qualquer das seguintes ações poderá ser acionada:

- Enviar um fluxo de tarefas, um job ou uma tarefa
- Responder a um prompt
- Executar comandos não Tivoli Workload Scheduler
- Registrar uma mensagem do operador
- Notificar usuários por meio de e-mail
- Enviar mensagens ao Tivoli Enterprise Console

Você também pode definir e executar regras de eventos que ajam na detecção da não conclusão, por um ou mais desses eventos ou por uma seqüência ou conjunto desses eventos, dentro de um período de tempo específico.

Informações adicionais estão disponíveis no Capítulo 7, “Automação de Carga de Trabalho Orientada pelo Evento em Execução”, na página 119.

Interfaces com o Usuário do Tivoli Workload Scheduler

Uma combinação de programas de interface gráfica, de linha de comandos e de API é fornecida para trabalhar com o Tivoli Workload Scheduler. Na verdade, a interface de linha de comandos está disponível para determinados recursos avançados que não estão disponíveis na interface gráfica com o usuário. Os programas da interface com o usuário do Tivoli Workload Scheduler disponíveis são:

Dynamic Workload Console

Uma interface com o usuário baseada na Web disponível para visualizar e controlar atividades de planejamento na produção em ambos os ambientes do Tivoli Workload Scheduler, distribuído e z/OS. Com o Dynamic Workload Console, você pode utilizar qualquer navegador suportado para acessar o ambiente do Tivoli Workload Scheduler a partir de qualquer local em sua rede.

É possível usar o Dynamic Workload Console para:

- Definir os objetos de planejamento no banco de dados do Tivoli Workload Scheduler
- Procurar e gerenciar objetos de planejamento envolvidos nas atividades atuais de planejamento
- Criar e controlar conexões com ambientes do Tivoli Workload Scheduler
- Enviar tarefas e fluxos de tarefas em produção
- Configurar preferências do usuário
- Criar e gerenciar regras de evento
- Definir e gerenciar tarefas críticas para a missão

Dynamic Workload Console deve ser instalado em um servidor que pode alcançar os nós do Tivoli Workload Scheduler utilizando conexões de rede. Consulte o guia Tivoli Workload Scheduler *Planejamento e Instalação* para obter informações.

composer

Um programa de linha de comandos utilizado para definir e gerenciar objetos de planejamento no banco de dados. Esse programa de interface e sua utilização são descritos no Capítulo 8, "Definindo Objetos no Banco de Dados", na página 137 e no Capítulo 9, "Gerenciando Objetos no Banco de Dados - Composer", na página 277.

conman

Um programa de linha de comandos utilizado para monitorar e controlar o processamento do plano de produção do Tivoli Workload Scheduler. Esse programa de interface é descrito no Capítulo 11, "Gerenciando Objetos no Plano - conman", na página 339.

Java™ API e plug-ins

Um conjunto de classes e métodos disponíveis em execução em um ambiente JAVA utilizado para criar sua interface customizada para gerenciar objetos de planejamento no banco de dados e no plano. Esta API não pode ser utilizada para criar sua interface customizada para configurar as opções globais. Além disso, é possível usar e modificar um conjunto de plug-ins que executam tarefas específicas ou criar seus próprios plug-ins. A API está disponível por meio de um Kit de Desenvolvimento de Software, que faz parte do produto. Para obter informações adicionais e aprender como acessar a documentação da API e dos plug-ins, consulte o *IBM Tivoli Workload Scheduler Developer's Guide: Software Development Kit (Integration Workbench)*.

optman

Um programa da linha de comandos usado para gerenciar as configurações que afetam o ambiente inteiro do Tivoli Workload Scheduler. Essas configurações, também chamadas de opções globais, são armazenadas no banco de dados. Este programa de interface é descrito no *Tivoli Workload Scheduler Administration Guide*.

planman
Um programa de linha de comandos utilizado para gerenciar a capacidade de planejamento do Tivoli Workload Scheduler. Esse programa de interface é descrito no "Linha de Comandos Planman" na página 82.

Interface de Serviços da Web
Uma interface que fornece um mecanismo de acesso baseado em serviços da web para um subconjunto de funcionalidades utilizadas para gerenciar tarefas e fluxos de tarefas no plano. Não permite gerenciar o plano, configurar opções globais, gerenciar objetos no banco de dados. Para obter informações adicionais, consulte o *IBM Tivoli Workload Scheduler Developer's Guide: Web Services*.

Você deve instalar o recurso Tivoli Workload Scheduler Command Line Client nos agentes tolerantes a falhas e sistemas fora da rede do Tivoli Workload Scheduler para utilizar os programas da linha de comandos **composer** e **optman** e para executar os comandos **planman showinfo** e **planman unlock**.

Para obter informações sobre como configurar as opções necessárias para permitir que um usuário acesse as interfaces de linha de comandos, consulte "Configurando Opções para Utilização das Interfaces com o Usuário" na página 53.

Iniciando a Produção

Esta seção fornece um caminho passo a passo de operações básicas que podem ser executadas rapidamente para implementar o Tivoli Workload Scheduler em seu ambiente, utilizando a interface de linha de comandos. Supõe-se que:

- Essas etapas sejam executadas no gerenciador de domínio principal imediatamente após a instalação bem-sucedida do produto nos sistemas onde você deseja executar suas atividades de planejamento.
- O ID do usuário utilizado para executar as operações é o mesmo que o utilizado para instalar o produto.

Se você não estiver familiarizado com o Tivoli Workload Scheduler, poderá seguir as etapas não opcionais para definir um número limitado de objetos de planejamento e incluir mais, à medida que se familiarizar com o produto. É possível iniciar, por exemplo, com dois ou três de seus aplicativos mais freqüentes, definindo objetos de planejamento para atender somente seus requisitos.

Como alternativa, é possível usar o Dynamic Workload Console para executar tarefas de modelagem e operacionais. Consulte a documentação do produto correspondente para obter informações adicionais.

A primeira atividade que você deve executar é acessar o banco de dados do Tivoli Workload Scheduler e definir o ambiente no qual deseja executar suas atividades de planejamento utilizando os tipos de objetos de planejamento do Tivoli Workload Scheduler. Para fazer isso, execute as seguintes etapas:

1. **Configure as variáveis de ambiente do Tivoli Workload Scheduler**
   Execute um dos seguintes scripts:
   `. ./`*`TWS_home`*`/tws_env.sh` para shells Bourne e Korn no UNIX
   `. ./`*`TWS_home`*`/tws_env.csh` para shells C no UNIX
   `TWS_home``\tws_env.cmd` no Windows
   em um shell de sistema para configurar as variáveis *PATH* e *TWS_TISDIR*.
2. **Conecte-se ao banco de dados do Tivoli Workload Scheduler**

Você pode utilizar a seguinte sintaxe para conectar-se ao gerenciador de domínio principal como *TWS_user*:

```
composer -user <TWS_user> -password <TWS_user_password>
```

em que *usuário_do_TWS* é o ID do usuário especificado no tempo de instalação.

Nota: Se quiser executar essa etapa e as seguintes a partir de um sistema diferente do gerenciador de domínio principal, você deve especificar os parâmetros de conexão ao iniciar o **composer**, conforme descrito em "Configurando Opções para Utilização das Interfaces com o Usuário" na página 53.

3. **Como opção, inclua no banco de dados as definições para descrever a topologia de seu ambiente de planejamento em termos de:**
   - **Domínios**

     Utilize essa etapa se quiser criar uma árvore hierárquica do caminho através do ambiente. A utilização de vários domínios reduz o tráfego de rede, reduzindo as comunicações entre o gerenciador de domínio principal e as outras estações de trabalho. Para obter informações adicionais, consulte "Definição de domínio" na página 154.
   - **Estações de trabalho**

     Defina uma estação de trabalho para cada máquina pertencente ao ambiente de planejamento, com exceção do gerenciador de domínio principal, que é definido automaticamente durante a instalação do Tivoli Workload Scheduler. Para obter informações adicionais, consulte "Definição de estação de trabalho" na página 139. O gerenciador de domínio principal é definido automaticamente no banco de dados na hora da instalação.
4. **Como opção, defina os usuários que podem executar tarefas nas estações de trabalho Windows**

   Defina qualquer usuário que possa executar tarefas utilizando o Tivoli Workload Scheduler, especificando o nome do usuário e a senha. Para obter informações adicionais, consulte "Definição do Usuário" na página 192.
5. **Como opção, defina calendários**

   Os calendários permitem que você determine se e quando uma tarefa ou um fluxo de tarefa deve ser executado. É possível utilizá-los para incluir ou excluir dias e horas para processamento. Os calendários não são estritamente necessários para definir os dias de planejamento para os fluxos de tarefa (os ciclos de execução `simple` ou `rule` também podem ser utilizados); sua principal meta é definir conjuntos de datas *globais* que podem ser reutilizados em vários fluxos de tarefa. Para obter informações adicionais, consulte "Definição do calendário" na página 197.
6. **Como opção, defina parâmetros, prompts e recursos**

   Para obter informações adicionais, consulte "Definição de Variável e Parâmetro" na página 198, "Definição do Prompt" na página 204 e "Definição do recurso" na página 206.
7. **Definir tarefas e fluxos de tarefa**

   Para obter informações adicionais, consulte "Tarefa" na página 711 e "Definição de fluxo de tarefas" na página 216.
8. **Como opção, defina as restrições e configurações para controlar quando as tarefas e os fluxos de tarefa são executados.**

   É possível definir dependências para tarefas e fluxos de tarefas. Pode haver até 40 dependências para um fluxo de tarefas. Podem ser:

- Dependências de recursos
- Dependências de arquivos
- Dependências follow de tarefa e de fluxo de tarefa
- Dependências de prompts

É possível definir as configurações de hora para tarefas e fluxo de tarefas para execução em termos de:

- Ciclos de Execução
- Restrições de hora

É possível ajustar a maneira como as tarefas são executadas simultaneamente em uma estação de trabalho ou em um fluxo de tarefa, configurando:

- Limite
- Prioridade

9. **Automatizar a extensão do plano no final do prazo de produção atual**

   Inclua o fluxo de tarefa `final` no banco de dados para desempenhar a extensão do plano de produção automática no final de cada prazo de produção atual executando o seguinte comando:

   ```
   add Sfinal
   ```

   Para obter informações adicionais, consulte "Automatizando o processamento do plano de produção" na página 101.

10. **Gere o plano**

    Execute o comando **JnextPlan** para gerar o plano de produção. Esse comando inicia o processamento das informações de planejamento armazenadas no banco de dados e cria o plano de produção para o espaço de tempo especificado no comando **JnextPlan**. O espaço de tempo padrão é 24 horas. Se você tiver automatizado a geração de plano conforme descrito na etapa anterior, é necessário executar somente o comando **JnextPlan** na primeira vez.

Ao concluir esse processo passo a passo, seu ambiente de planejamento estará ativado e em execução, com o processamento de batch de uma seqüência ordenada de tarefas e fluxos de tarefas sendo executado com recursos definidos em um conjunto de estações de trabalho, se definido. Por padrão, na primeira vez que o comando **JnextPlan** for executado, o número de tarefas que poderão ser executadas simultaneamente em uma estação de trabalho será zero, portanto, certifique-se de aumentar esse valor alterando `limit cpu` para permitir a execução da tarefa nessa estação de trabalho. Consulte a seção "limit cpu" na página 384 para obter detalhes adicionais.

Se quiser modificar algo enquanto o plano de produção já está em processo, utilize o programa **conman**. Enquanto o plano de produção está processando na rede, ainda é possível continuar a definir ou modificar tarefas e fluxos de tarefas no banco de dados. Considere, no entanto, que essas modificações serão utilizadas somente se você enviar as tarefas modificadas ou os fluxos de tarefas utilizando o comando **sbj** para tarefas ou **sbs** para fluxos de tarefas, em uma estação de trabalho que já recebeu o plano ou após um novo plano de produção ser gerado, utilizando **JnextPlan**. Consulte o Capítulo 11, "Gerenciando Objetos no Plano - conman", na página 339 para obter detalhes sobre o programa **conman** e as operações que é possível executar no plano de produção em processo.

Capítulo 2. Entendendo Processos e Comandos Básicos

Em uma rede multicamada do Tivoli Workload Scheduler, localmente, em cada estação de trabalho, um grupo de processos de planejamento especializados executa o gerenciamento de tarefa e envia as informações de volta sobre o processamento de tarefa em toda a árvore hierárquica até ogerenciador de domínio principal ser alcançado. Usando as informações recebidas das estações de trabalho, o gerenciador de domínio principal então atualiza sua cópia do arquivo `Symphony` e o plano replicado no banco de dados, que contêm os registros que descrevem as atividades de processamento de tarefa a serem executadas na rede do Tivoli Workload Scheduler durante o plano de produção atual, e envia as atualizações nas atividades a serem executadas nas estações de trabalho envolvidas.

Emitindo Comandos em Sistemas Operacionais Windows

Em sistemas operacionais Windows, assegure-se de estar emitindo os comandos do Tivoli Workload Scheduler a partir de um prompt de comandos com o nível de privilégio **Executar como administrador**.

Processos da Estação de Trabalho do Tivoli Workload Scheduler

O gerenciamento de comunicação entre estações de trabalho e processamento de tarefa local, juntamente com a notificação de atualizações de estado, são executados em cada estação de trabalho do Tivoli Workload Scheduler por uma série de processos de gerenciamento que estão ativos enquanto o mecanismo está em execução. Nos agente tolerante a falhass e gerenciadores de domínio, esses processos são baseados na infraestrutura do WebSphere Application Server. Essa infra-estrutura é instalada automaticamente com a estação de trabalho e permite que o Tivoli Workload Scheduler:

- Comunique-se através da rede do Tivoli Workload Scheduler.
- Gerencie os mecanismos de autenticação de clientes remotos, como programas de linha de comandos, conectando-se aogerenciador de domínio principal usando protocolos HTTP ou HTTPS.

Para obter informações sobre como iniciar e parar a infraestrutura do WebSphere Application Server e os processos do Tivoli Workload Scheduler, em uma estação de trabalho consulte “Iniciando e Parando Processos em uma Estação de Trabalho” na página 37. Exceto para iniciar e parar manualmente oWebSphere Application Server e gerenciar os parâmetros de conexão ao se comunicar na rede do Tivoli Workload Scheduler, a infraestrutura do WebSphere Application Server é transparente ao usar oTivoli Workload Scheduler.

Neste guia, os *processos do Tivoli Workload Scheduler* ou *processos da estação de trabalho* são usados para identificar os seguintes processos:

netman
monman
writer
mailman
batchman
jobman

© Copyright IBM Corp. 1999, 2014

Com exceção dos agentes padrão, esses processos são iniciados na seguinte ordem nas estações de trabalho do Tivoli Workload Scheduler:

netman

Netman é o processo de Gerenciamento de Redes. É iniciado pelo comando **Startup** e comporta-se como um programa listener que recebe pedidos para iniciar, parar, vincular ou desvincular a rede. **Netman** examina cada solicitação recebida e cria um processo local do Tivoli Workload Scheduler.

monman

Monman é um processo iniciado por **netman** e é usado em gerenciamento de eventos. Ele inicia serviços **ssmagent** e de monitoramento que têm a tarefa de detectar os eventos definidos nas regras de eventos implementadas e ativadas na estação de trabalho específica. Quando esses serviços capturam quaisquer desses eventos, após uma ação de filtragem preliminar, eles os enviam para o servidor de processamento de eventos que é normalmente executado no gerenciador de domínio principal. Se nenhuma configuração de regra de evento for transferida por download para a estação de trabalho, os serviços de monitoramento permanecem inativos.

O processo de comunicação entre os agentes de monitoramento e o servidor de processamento de eventos é independente da topologia de rede do Tivoli Workload Scheduler. Ele se baseia diretamente no número da porta EIF do processador de eventos e as informações de evento fluem diretamente dos agentes de monitoramento sem passar pelos gerenciadores de domínio intermediários. Um grau de tolerância a falhas é garantido pelas memórias cache locais, que armazenam temporariamente as ocorrências de evento nos agentes caso a comunicação com o processador de eventos esteja inativa.

writer **Writer** é um processo iniciado por **netman** para transmitir mensagens que chegam para o processo **mailman** local. Os processos **writer** (pode existir mais de um em uma estação de trabalho de gerenciador de domínio) são iniciados por pedidos para vincular (consulte “link” na página 386) e são parados por pedidos para desvincular (consulte “unlink” na página 466) (ou quando o**mailman** de comunicação é encerrado).

mailman

Mailman é o processo de Gerenciamento de Correio. Ele roteia mensagens para estações de trabalho local ou remota. Em um gerenciador de domínio, processos **mailman** adicionais podem ser criados para dividir o carregamento do **mailman** devido à inicialização dos agentes e para aprimorar a conveniência das mensagens. Quando o gerenciador de domínio é iniciado, ele cria uma instância de processo **mailman** separada para cada *ServerID* especificado nas definições da estação de trabalho dos agente tolerante a falhass e dos agentes padrão que ele gerencia. Cada estação de trabalho é contatada por seu próprio*ServerID* no gerenciador de domínio. Para obter informações adicionais, consulte “Definição de estação de trabalho” na página 139.

batchman

Batchman é o processo de Controle de Produção. Ele interage diretamente com a cópia do arquivo `Symphony` distribuída para as estações de trabalho no início do período de produção e a atualiza. **Batchman** executa várias funções:

- Gerencia localmente o processamento e a atualização de planos.

- Resolve dependências de tarefas e de fluxos de tarefas.
- Seleciona tarefas a serem executadas.
- Atualiza o plano com os resultados de processamento de tarefa.

Batchman é o único processo que pode atualizar o arquivo `Symphony`.

jobman

Jobman é o processo de Gerenciamento de Job. Ativa tarefas sob a direção de **batchman** e relata o status da tarefa de volta ao**mailman**. É responsável por controlar os estados da tarefa e pela configuração do ambiente conforme definido nos scripts `jobmanrc` e `.jobmanrc` quando solicitada a ativação de tarefas. Para obter informações sobre esses scripts, consulte Capítulo 3, "Configurando o Ambiente de Tarefa", na página 45. Quando o processo **jobman** recebe uma mensagem de tarefa de ativação do **batchman**, ele cria um processo de monitoramento de tarefa. O número máximo de processos de monitoramento de tarefa que podem ser criados em uma estação de trabalho é definido utilizando o comando **limit cpu** a partir do prompt de linha de comandos **conman** (consulte "limit cpu" na página 384).

job monitor (jobman em UNIX, JOBMON.exe e joblnch.exe no sistema operacional Windows)

O processo de monitoramento de tarefa executa primeiramente um conjunto de ações que definem o ambiente antes que a tarefa seja ativada e, em seguida, ativa a tarefa executando o arquivo de script ou o comando especificado na definição de tarefa. Para obter detalhes adicionais sobre como especificar o arquivo de script ou o comando ativado com o job, consulte "Tarefa" na página 711.

As atividades de configuração consistem na ativação do arquivo de configuração padrão (*TWS_home*`/jobmanrc` no UNIX e *TWS_home*`/jobmanrc.cmd` no Windows) que contém configurações que se aplicam a todas as tarefas em execução na estação de trabalho. Além disso, nas estações de trabalhoUNIX, um script de configuração local *TWS_user*`/.jobmanrc` é ativado, se ele existir no diretório inicial do usuário que está ativando a tarefa. Esse arquivo de configuração local contém configurações que se aplicam somente a jobs ativados pelo usuário específico. Se qualquer uma dessas etapas falhar, a tarefa termina no estado FAIL.

Atenção: Se, em sistemas Windows, existir uma variável do sistema chamada `TEMP`, o usuário *TWS_user* deve estar autorizado para criar arquivos no diretório no qual a variável é configurada. Se esse requisito não for atendido, o arquivo binário**JOBMON.exe** falha em ser iniciado com sucesso.

Todos os processos, exceto**jobman**, são executados como *TWS_user*. **Jobman** é executado como raiz.

Em estações de trabalho de agente padrão, o processo **batchman** não é ativado porque esse tipo de estação de trabalho não gerencia o planejamento de job. Essas estações de trabalho somente ativam jobs na direção de seu gerenciador de domínio. Localmente, na estação de trabalho, os processos de gerenciamento aguardam um pedido para ativar um job no gerenciador de domínio no modo LISTEN. Quando o pedido é recebido, o job é ativado localmente e o resultado é

retornado para o gerenciador de domínio. Para obter informações adicionais sobre estações de trabalho de agente padrão, consulte o *IBM Tivoli Workload Scheduler: Guia de Planejamento e Instalação*.

O Figura 3 na página 35 mostra a árvore de processo nas estações de trabalho do Tivoli Workload Scheduler, diferente dos agentes padrão, instalados no UNIX:

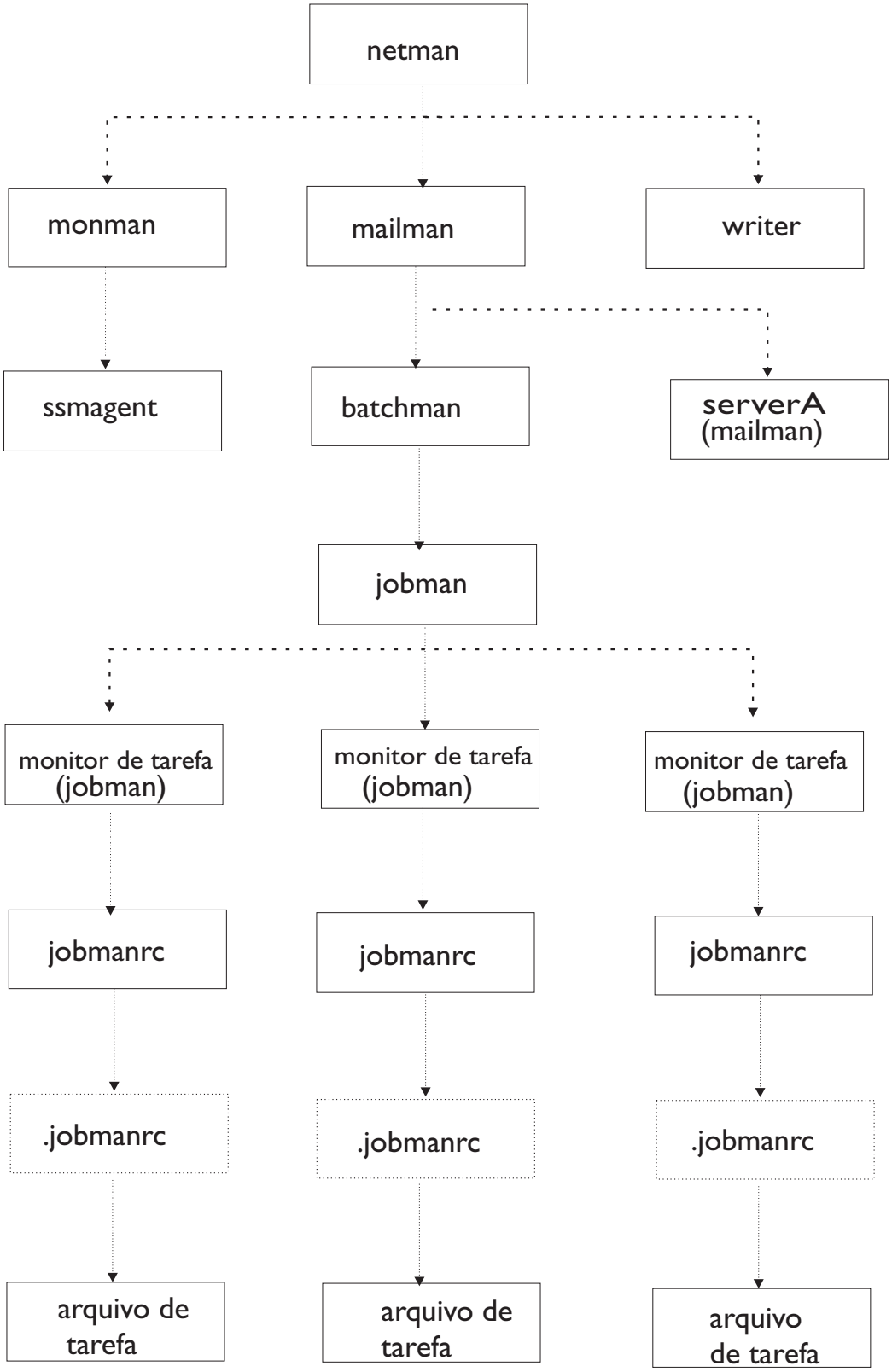


Figura 3. Árvore de Processos no UNIX

Figura 4 na página 36 mostra a árvore de processo nas estações de trabalhoTivoli Workload Scheduler, diferente dos agentes padrão, instalados no Windows:

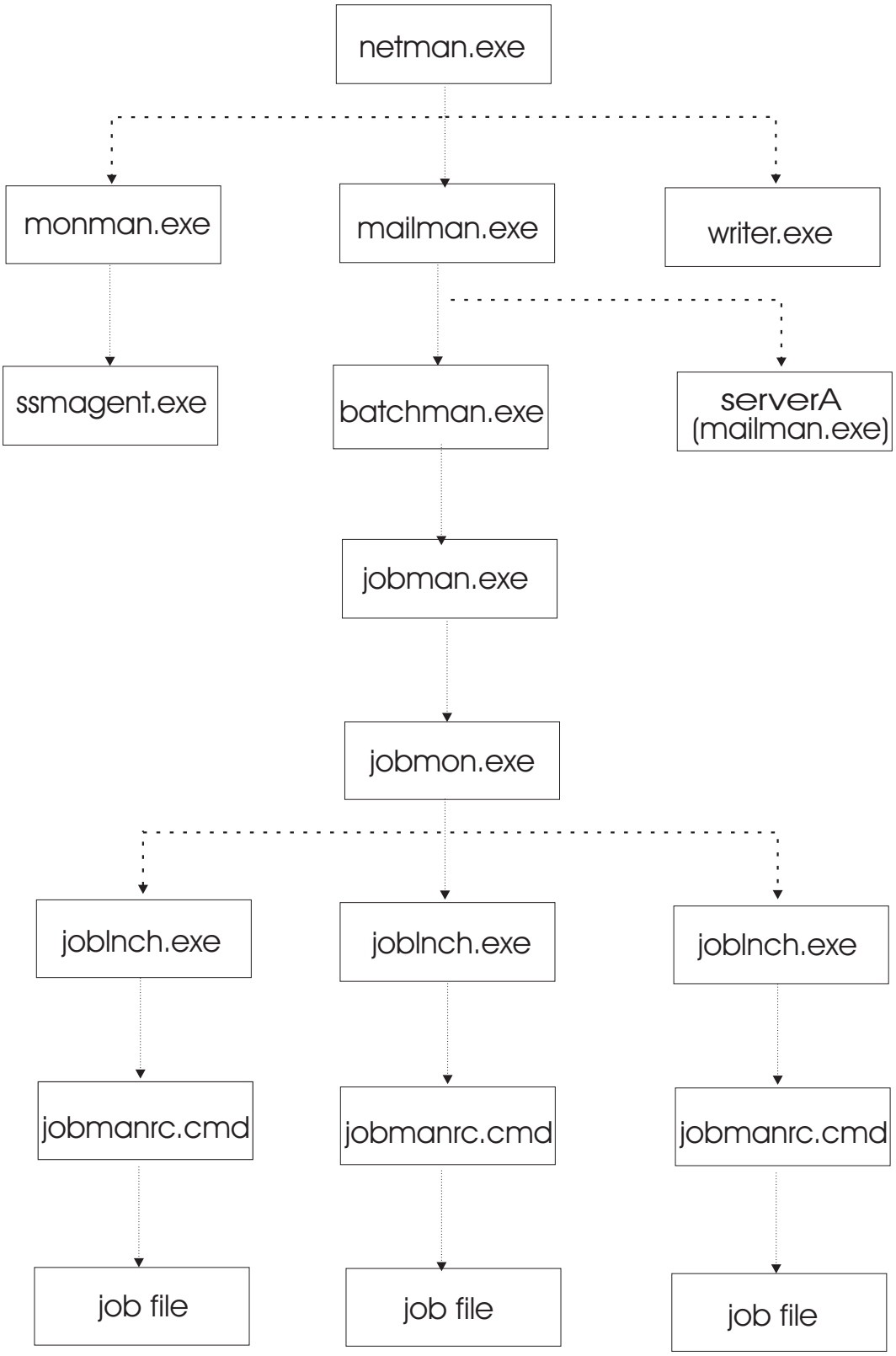


Figura 4. Árvore de Processos no Windows

Em plataformas Windows, há um serviço adicional, o `Tivoli Token Service`, que permite que os processos Tivoli Workload Scheduler sejam ativados como se fossem emitidos pelo usuário Tivoli Workload Scheduler.

Iniciando e Parando Processos em uma Estação de Trabalho

O tipo do sistema operacional instalado na estação de trabalho determina como os processos do Tivoli Workload Scheduler podem ser iniciados a partir da linha de comandos. A Tabela 6 explica como é possível iniciar e parar a infra-estrutura do WebSphere Application Server e os processos do Tivoli Workload Scheduler em uma estação de trabalho com base no sistema operacional instalado.

Tabela 6. Iniciando e Parando oTivoli Workload Scheduler em uma Estação de Trabalho

Ação	Comandos utilizados na plataforma UNIX	Comandos utilizados na plataforma Windows
Iniciar todos os processos do Tivoli Workload Scheduler, incluindoWebSphere Application Server e o mecanismo de monitoramento de eventos.	conman start conman startappserver conman startmon	conman start conman startappserver conman startmon
Iniciar o netman e o WebSphere Application Server. No Windows, inicia também o`Tivoli Token Service`	./StartUp.sh	StartUp
Parar todos os processos do Tivoli Workload Scheduler e o WebSphere Application Server.	conman shutdown ./stopWas.sh	conman shutdown -appsrv shutdown -appsrv
Parar todos os processos do Tivoli Workload Scheduler com a exceção do WebSphere Application Server.	conman shutdown	conman shutdown shutdown
Iniciar todos os processos do Tivoli Workload Scheduler com a exceção do WebSphere Application Server e do mecanismo de monitoramento de eventos.	conman start	conman start
Pare todos os processos Tivoli Workload Scheduler, exceto**netman**, **monman**, **writer** e **appservman**.	parada do conman	parada do conman
Parar todos os processos do Tivoli Workload Scheduler (incluindo**netman**).	conman shutdown	conman shutdown shutdown
Iniciar oWebSphere Application Server	./startWas.sh ou conman startappserver	startWas ou conman startappserver
Parar oWebSphere Application Server	./stopWas.sh ou conman stopappserver	stopWas ou conman stopappserver
Iniciar o mecanismo de monitoramento de eventos	conman startmon	conman startmon
Parar o mecanismo de monitoramento de eventos	conman stopmon	conman stopmon
Iniciar oagent localmente	./StartUpLwa.sh **Nota:** pode ser executado somente por *TWS_user* ou pelo usuário root.	startuplwa **Nota:** No Windows 2008, deve ser executado como Administrador.
Parar oagent localmente	./ShutDownLwa.sh **Nota:** pode ser executado somente por *TWS_user* ou pelo usuário root.	shutdownlwa **Nota:** No Windows 2008, deve ser executado como Administrador.

Nota: Nos sistema Windows, não utilize os serviços do Windows para parar o WebSphere Application Server. Em vez disso, utilize um dos comandos listados nesta tabela. Se você utilizar esses serviços do Windows para parar o WebSphere Application Server, o processo *appserverman*, que continua sendo executado, o iniciará novamente.

Consulte “StartUp” na página 527 para obter informações adicionais sobre o comando utilitário **StartUp**.

Consulte “shutdown” na página 527 para obter informações adicionais sobre o comando utilitário **shutdown**.

Consulte o *IBM Tivoli Workload Scheduler: Guia de Administração* para obter mais informações sobre os comandos **startWas** e **stopWas**.

Consulte "start" na página 437 para obter informações adicionais sobre o comando **conman start**.

Consulte "stop" na página 442 para obter informações adicionais sobre o comando **conman stop**.

Consulte "shutdown" na página 436 para obter informações adicionais sobre o comando **conman shutdown**.

Consulte "startappserver" na página 439 para obter informações adicionais sobre o comando **conman startappserver**.

Consulte "stopappserver" na página 445 para obter informações adicionais sobre o comando **conman stopappserver**.

Consulte "startmon" na página 441 para obter informações adicionais sobre o comando **conman startmon**.

Consulte "stopmon" na página 448 para obter informações adicionais sobre o comando **conman stopmon**.

Se o agente estiver instalado em um sistema Windows, o WebSphere Application Server e os processos **netman** serão iniciados automaticamente no horário de início dos serviços juntamente com o `Tivoli Token Service`. Se o agente estiver instalado em um sistema UNIX, o WebSphere Application Server e os processos **netman** poderão ser iniciados automaticamente no horário de início incluindo uma instrução que chama **Startup** no arquivo `/etc/inittab`.

Iniciando e parando oagent

O tipo de sistema operacional instalado na estação de trabalho determina como os agentes podem ser iniciados a partir da linha de comandos.

Tabela 7. Iniciando e parando oagent

Ação	Comandos utilizados na plataforma UNIX	Comandos utilizados na plataforma Windows
Iniciar oagent localmente	./StartUpLwa.sh **Nota:** pode ser executado somente por *TWS_user* ou pelo usuário root.	startuplwa **Nota:** No Windows 2008, deve ser executado como Administrador.
Parar oagent localmente	./ShutDownLwa.sh **Nota:** pode ser executado somente por *TWS_user* ou pelo usuário root.	shutdownlwa **Nota:** No Windows 2008, deve ser executado como Administrador.

Para obter informações adicionais sobre como parar e iniciar oagent, consulte ShutDownLwa e StartUpLwa.

Comunicação Interprocesso de Estação de Trabalho

O Tivoli Workload Scheduler utiliza filas de mensagens para a comunicação local entre os processos. Há quatro arquivos de mensagens que residem no diretório *TWS_home*:

NetReq.msg
Esse arquivo de mensagens é lido pelo processo **netman** para comandos locais. Ele recebe mensagens como START, STOP, LINK e UNLINK.

Mailbox.msg
Esse arquivo de mensagens é lido pelo processo **mailman**. Recebe mensagens, por meio da interface gráfica com o usuário (Dynamic Workload Console) ou do gerenciador de console (**conman**), recebidas dos processos **batchman** e **jobman** locais e de outras estações de trabalho do Tivoli Workload Scheduler na rede.

Intercom.msg
Esse arquivo de mensagens é lido pelo processo **batchman** e contém instruções enviadas pelo processo local **mailman**.

Courier.msg
Esse arquivo de mensagens é gravado pelo processo **batchman** e lido pelo processo **jobman**.

PlanBox.msg
Esse arquivo de mensagens é gravado pelo processo **batchman** e lido pelo mecanismo.

Server.msg
Esse arquivo de mensagens é gravado pelo processo **batchman** e lido pelo mecanismo.

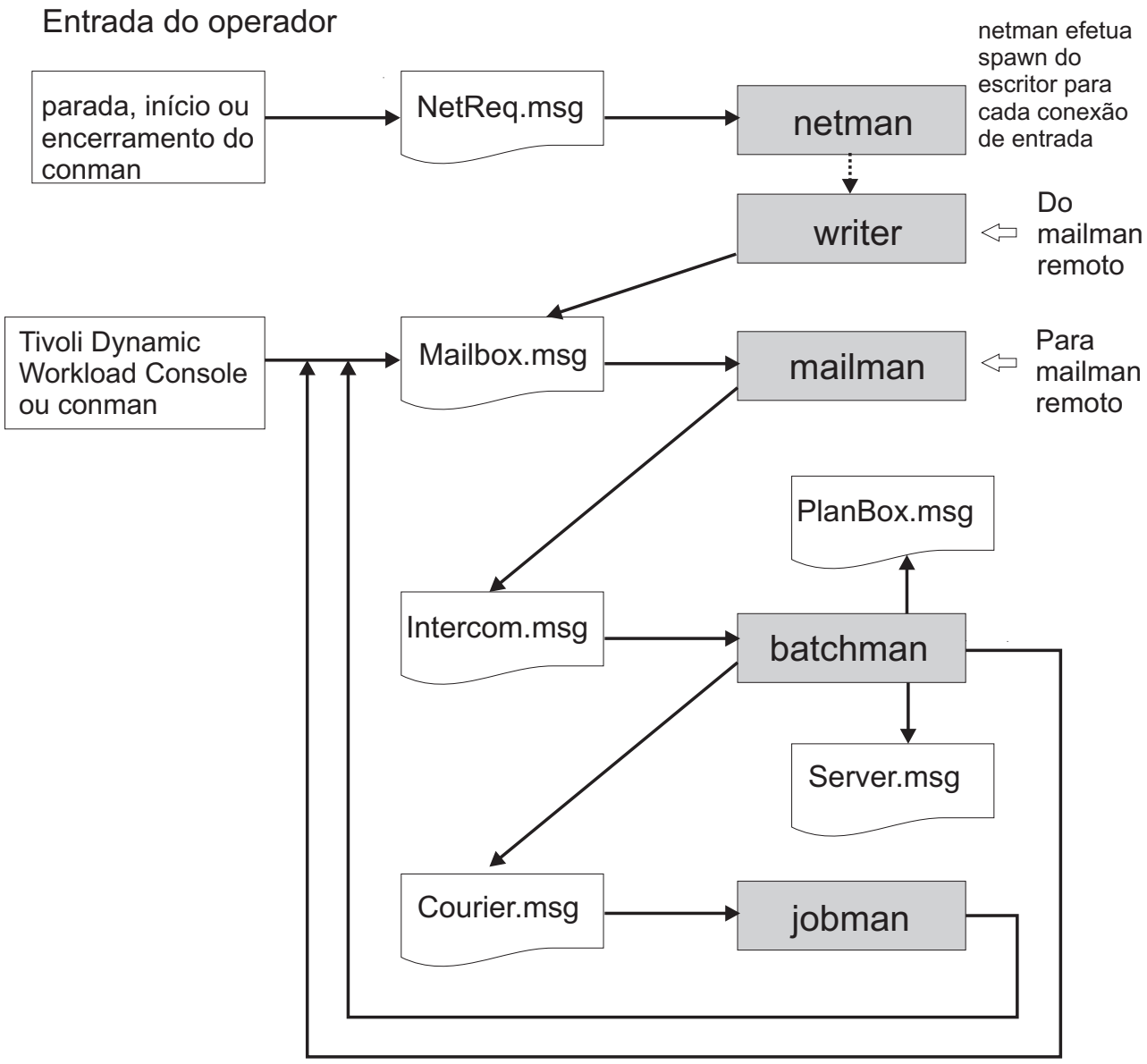


Figura 5. Comunicação entre os Processos

Esses arquivos possuem um tamanho máximo padrão de 10 MB. O tamanho máximo pode ser alterado usando o utilitário **evtsize** (consulte "evtsize" na página 509).

Comunicação da Rede do Tivoli Workload Scheduler

O Tivoli Workload Scheduler utiliza o protocolo TCP/IP para comunicação de rede. O nome de nó e o número da porta usados para estabelecer a conexão TCP/IP são configurados para cada estação de trabalho em sua definição da estação de trabalho. Consulte o "Definição de estação de trabalho" na página 139 para obter detalhes adicionais.

Uma tecnologia *armazenar e encaminhar* é usada pelo Tivoli Workload Scheduler para manter consistência e tolerância a falhas a todo tempo pela rede, enfileirando mensagens nos arquivos de mensagens enquanto a conexão não está ativa. Quando a comunicação TCP/IP é estabelecida entre sistemas, o Tivoli Workload Scheduler

fornece comunicação bidirecional entre estações de trabalho usando links. Os links são controlados pelo sinalizador **autolink** configurado em "Definição de estação de trabalho" na página 139 e pelos comandos "link" na página 386 e "unlink" na página 466 emitidos a partir do programa de linha de comandos **conman**.

Quando um link é aberto, as mensagens são transmitidas entre duas estações de trabalho. Quando um link é fechado, a estação de trabalho de envio armazena mensagens em um arquivo de mensagens local e os envia para a estação de trabalho de destino quando o link é reaberto.

Existem basicamente dois tipos de comunicação que ocorrem no ambiente do Tivoli Workload Scheduler, inicialização de conexão e entrega de eventos de planejamento na forma de mensagens de alteração de estado durante o período de processamento. Esses dois tipos de comunicação são explicados agora em detalhes.

Configuração de Inicialização de Conexão e de Comunicação Bidirecional

Estas são as etapas envolvidas no estabelecimento de um link bidirecional do Tivoli Workload Scheduler entre um gerenciador de domínio e um agente tolerante a falhas remoto:

1. No gerenciador de domínio, o processo **mailman** lê o nome do host, endereço TCP/IP e número de porta do agente tolerante a falhas a partir do arquivo `Symphony`.
2. O processo **mailman** no gerenciador de domínio estabelece uma conexão TCP/IP ao processo **netman** no agente tolerante a falhas usando as informações obtidas do arquivo `Symphony`.
3. O processo **netman** no agente tolerante a falhas determina que a solicitação está vindo do processo **mailman** no gerenciador de domínio e cria um novo processo **writer** para manipular a conexão recebida.
4. O processo **mailman** no gerenciador de domínio agora está conectado ao processo **writer** no agente tolerante a falhas. O processo **writer** no agente tolerante a falhas comunica o número de execução atual de sua cópia do arquivo `Symphony` para o processo **mailman** no gerenciador de domínio. Esse número de execução é o identificador utilizado pelo Tivoli Workload Scheduler para reconhecer cada arquivo `Symphony` gerado pelo **JnextPlan**. Essa etapa é necessária para o gerenciador de domínio verificar se o plano atual já foi enviado aoagente tolerante a falhas.
5. O processo **mailman** no gerenciador de domínio compara o número de execução de seu arquivo `Symphony` com o número de execução do arquivo `Symphony` no agente tolerante a falhas. Se os números de execução forem diferentes, o processo **mailman** no gerenciador de domínio envia ao processo **writer** no agente tolerante a falhas a cópia mais recente do arquivo `Symphony`.
6. Quando o arquivo `Symphony` atual está no local no agente tolerante a falhas, o processo **mailman** no gerenciador de domínio envia um comando `start` aoagente tolerante a falhas.
7. O processo **netman** no agente tolerante a falhas inicia o processo **mailman** local. Nesse ponto, um link de comunicação unidirecional é estabelecido a partir do gerenciador de domínio com o agente tolerante a falhas.
8. O processo **mailman** no agente tolerante a falhas lê o nome do host, endereço TCP/IP e número da porta do gerenciador de domínio a partir do arquivo `Symphony` e os utiliza para estabelecer o uplink de volta para o processo **netman** no gerenciador de domínio.

9. O processo **netman** no gerenciador de domínio determina que a solicitação está vindo do processo **mailman** no agente tolerante a falhas e cria um novo processo **writer** para manipular a conexão recebida. O processo **mailman** no agente tolerante a falhas é agora conectado ao**writer** no gerenciador de domínio e um link de comunicação bidirecional é estabelecido. Consequentemente, o processo **writer** no gerenciador de domínio grava mensagens recebidas do agente tolerante a falhas no arquivo `Mailbox.msg` no gerenciador de domínio, e o processo **writer** no agente tolerante a falhas grava as mensagens do gerenciador de domínio no arquivo `Mailbox.msg` no agente tolerante a falhas.

O evento de processamento e planejamento de tarefa são fornecidos em forma de mensagens de mudança de estado durante o dia de processamento executado localmente pelo agente tolerante a falhas

Durante o período de produção, o arquivo `Symphony` presente no agente tolerante a falhas é lido e atualizado com informações de mudança de estado sobre tarefas que são executadas localmente pelos processos da estação de trabalho do Tivoli Workload Scheduler. Essas são as etapas executadas localmente no agente tolerante a falhas para leitura e atualização do arquivo `Symphony` e para processar tarefas:

1. O processo **batchman** lê um registro no arquivo `Symphony` que afirma que `job1` deve ser ativado na estação de trabalho.
2. O processo **batchman** grava no arquivo `Courier.msg` que `job1` precisa ser iniciado.
3. O processo **jobman** lê essas informações no arquivo `Courier.msg`, inicia `job1` e grava no arquivo `Mailbox.msg` que `job1` foi iniciado com seu *process_id* e *timestamp*.
4. O processo **mailman** lê estas informações em seu arquivo `Mailbox.msg` e envia uma mensagem que informa que `job1` iniciou com seu *process_id* e *timestamp*, para o arquivo `Mailbox.msg` no gerenciador de domínio e o arquivo local `Intercom.msg` no agente tolerante a falhas.
5. O processo **batchman** no agente tolerante a falhas lê a mensagem no arquivo `Intercom.msg` e atualiza a cópia local do arquivo Symphony.
6. Quando o job `job1` conclui o processamento, o processo **jobman** atualiza o arquivo `Mailbox.msg` com as informações de que `job1` foi concluído.
7. O processo **mailman** lê as informações no arquivo `Mailbox.msg` e envia uma mensagem que o `job1` concluiu para o arquivo `Mailbox.msg` no gerenciador de domínio e o arquivo `Intercom.msg` local no agente tolerante a falhas.
8. O processo **batchman** no agente tolerante a falhas lê a mensagem no arquivo `Intercom.msg`, atualiza a cópia local no arquivo `Symphony` e determina a próxima tarefa a ser executada.

Para obter informações sobre como ajustar o processamento de tarefa em uma estação de trabalho, consulte o *IBM Tivoli Workload Scheduler: Guia de Administração*.

Suporte para o Protocolo da Internet Versão 6

O Tivoli Workload Scheduler suporta o Protocolo da Internet versão 6 (IPv6) além do IPv4 legado. Para ajudar os clientes no estágio da transição de um ambiente IPv4 para um ambiente IPv6 completo, o Tivoli Workload Scheduler fornece

suporte de pilha dupla IP. Em outros termos, o produto foi projetado para comunicação utilizando endereços IPv4 e IPv6 simultaneamente com outros aplicativos utilizando IPv4 ou IPv6.

Para este fim, as funções `gethostbyname` e `gethostbyaddr` foram eliminadas do Tivoli Workload Scheduler já que elas suportam exclusivamente o IPv4. Elas foram substituídas pela nova API `getaddrinfo` que torna o mecanismo cliente-servidor inteiramente independente de protocolo.

A função `getaddrinfo` manipula a conversão de nome-para-endereço e de serviço-para-porta e retorna as estruturas `sockaddr` em vez de uma lista de endereços. Estas estruturas `sockaddr` podem, então, ser utilizadas pelas funções de soquete diretamente. Desta maneira, `getaddrinfo` oculta todas as dependências de protocolo na função de biblioteca, à qual elas pertencem. O aplicativo lida somente com as estruturas de endereço de soquete que são preenchidas por `getaddrinfo`.

Capítulo 3. Configurando o Ambiente de Tarefa

Este capítulo descreve como customizar a maneira que o gerenciamento de tarefa é executado em uma estação de trabalho. Esta customização é feita designando-se localmente em cada estação de trabalho valores a variáveis que têm um impacto no processamento de **jobman**. Este capítulo inclui as seguintes seções:

- "Visão Geral do Ambiente de Tarefa"
- "Variáveis de Ambiente Exportadas por jobman" na página 46
- "Customizando o Processamento da Tarefa em uma Estação de Trabalho UNIX - jobmanrc" na página 48
- "Customizando o Processamento de Tarefas para um Usuário nas Estações de Trabalho UNIX - .jobmanrc" na página 49
- "Customizando o Processamento de Tarefa em uma Estação de Trabalho Windows - jobmanrc.cmd" na página 51
- "Customizando Processamento de Tarefa em uma Estação de Trabalho com o Windows - djobmanrc.cmd" na página 52

Visão Geral do Ambiente de Tarefa

Em cada estação de trabalho, as tarefas são ativadas pelo processo de controle de produção **batchman**. O processo **batchman** resolve todas as dependências de tarefas para assegurar a ordem correta do processamento da tarefa e, em seguida, enfileira uma mensagem de ativação da tarefa para o processo **jobman**.

Cada um dos processos ativados por **jobman**, incluindo os scripts e as tarefas de configuração, retém o nome do usuário registrado com o logon da tarefa. Tarefas enviadas (tarefas, arquivos ou comandos enviados não através de um planejado, mas *manualmente* por um usuário) mantêm o nome do usuário que enviou.

O processo **jobman** inicia um processo de monitor de tarefa que começa configurando um grupo de variáveis de ambiente e, em seguida, executa um script de configuração padrão denominado *TWS_home*/**jobmanrc**, que pode ser customizado. O script **jobmanrc** configura variáveis que são usadas para configurar localmente na estação de trabalho a maneira como as tarefas são ativadas, independentemente do usuário.

Nas estações de trabalho UNIX, se o usuário tiver permissão para utilizar um script de configuração local e o script *USER_HOME*/**.jobmanrc** existir, o script de configuração local **.jobmanrc** será *também* executado. Em seguida, o job é executado pelo script de configuração padrão ou pelo local. Os resultados do processamento da tarefa são relatados para **jobman**, que, por sua vez, atualiza o arquivo `Mailbox.msg` com as informações sobre o status de conclusão. Para que os jobs sejam executados com o ambiente do usuário, inclua a seguinte instrução no script de configuração local:

```
. $USER_home/.profile
```

Nota: Antes de incluir o `.profile` no arquivo `.jobmanrc`, certifique-se de que ele não contenha nenhuma configuração *stty* ou nenhuma etapa que precise da intervenção manual do usuário. Caso haja, inclua no arquivo `.jobmanrc` somente as etapas necessárias contidas no `.profile`.

© Copyright IBM Corp. 1999, 2014

Em estações de trabalho com o Windows, o script de configuração local djobmanrc.cmd é executado se existir no diretório Documentos e Configurações do usuário, que é representado pela variável de ambiente %USERPROFILE% e depende da instalação do idioma do Windows. O script djobmanrc.cmd será executado pelo script jobmanrc.cmd.

Variáveis de Ambiente Exportadas por jobman

As variáveis listadas na Tabela 8 são configuradas localmente na estação de trabalho e exportadas por **jobman** nos sistemas operacionais Windows:

Tabela 8. Variáveis de Ambiente de Job para o Windows

Nome da Variável	Valor
COMPUTERNAME	O valor de *COMPUTERNAME* definido no ambiente do usuário.
HOME	O caminho no qual a instância do Tivoli Workload Scheduler foi instalada.
HOMEDRIVE	O valor de *HOMEDRIVE* definido no ambiente do usuário.
HOMEPATH	O valor de *HOMEPATH* definido no ambiente do usuário.
LANG	O valor de *LANG* definido no ambiente do usuário. Se não estiver definido, seu valor será definido como C.
LOGNAME	O nome do usuário de login.
MAESTRO_OUTPUT_STYLE	A configuração para o estilo de saída para nomes longos de objetos.
SystemDrive	O valor de *SYSTEMDRIVE* definido no ambiente do usuário.
SystemRoot	O valor de *SYSTEMROOT* definido no ambiente do usuário.
TEMP	O valor de *TEMP* definido no ambiente do usuário. Se não for especificado, seu valor será definido como `c:\temp`.
TIVOLI_JOB_DATE	A data planejada para um job.
TMPTEMP	O valor de *TMP* definido no ambiente do usuário. Se não for especificado, seu valor será definido como `c:\temp`.
TMPDIR	O valor de *TMPDIR* definido no ambiente do usuário. Se não for especificado, seu valor será definido como `c:\temp`.
TWS_PROMOTED_JOB	Aplica-se às funções de Garantia de Serviço da Carga de Trabalho. Pode ser `YES` ou `No` . Quando o valor é `YES`, significa que a tarefa (uma tarefa crítica ou uma de suas predecessoras) foi promovida.
TZ	O fuso horário, se ele foi configurado no ambiente do sistema operacional.
UNISON_CPU	O nome desta estação de trabalho.
UNISON_DIR	O valor de *UNISON_DIR* definido no ambiente do usuário.
UNISON_EXEC_PATH	O caminho completo de `jobmanrc`.
UNISONHOME	O caminho no qual a instância do Tivoli Workload Scheduler foi instalada.
UNISON_HOST	O nome da CPU do host.
UNISON_JOB	O identificador de tarefa absoluto: `worktation#sched_id.job` .
UNISON_JOBNUM	O número do job.
UNISON_MASTER	O nome da estação de trabalho principal.
UNISON_RUN	O número de execução da produção atual do Tivoli Workload Scheduler.
UNISON_SCHED	O nome do fluxo de tarefas.
UNISON_SCHED_DATE	A data de produção (aammdd) do Tivoli Workload Scheduler relatada no cabeçalho do arquivo `Symphony`.
UNISON_SCHED_ID	O *jobstreamID* do fluxo de tarefas que contém a tarefa no processo.
UNISON_SCHED_IA	O *StartTime* do fluxo de tarefas que contém a tarefa no processo.
UNISON_SCHED_EPOCH	A data de produção do Tivoli Workload Scheduler expressa no formato de época.
UNISON_SHELL	O shell de login do usuário executando a tarefa.
UNISON_STDLIST	O nome do caminho do arquivo de lista padrão do job.
UNISON_SYM	O número do registro `Symphony` da tarefa ativada.
USERDOMAIN	O valor de *USERDOMAIN* definido no ambiente do usuário.
USERNAME	O valor de *USERNAME* definido no ambiente do usuário.
USERPROFILE	O valor de *USERPROFILE* definido no ambiente do usuário.

As variáveis listadas na Tabela 9 na página 47 são configuradas localmente na estação de trabalho e exportadas por **jobman** nos sistemas operacionais UNIX:

Tabela 9. Variáveis de Ambiente de Job para o UNIX

Nome da Variável	Valor
HOME	O diretório home do usuário.
LANG	O valor de *LANG* definido no ambiente do usuário.
LD_LIBRARY_PATH	O valor de *LD_LIBRARY_PATH* definido no ambiente do usuário.
LD_RUN_PATH	O valor de *LD_LIBRARY_PATH* definido no ambiente do usuário.
LOGNAME	O nome do usuário de login.
MAESTRO_OUTPUT_STYLE	A configuração para o estilo de saída para nomes longos de objetos.
PATH	`/bin:/usr/bin`
TIVOLI_JOB_DATE	A data planejada para um job.
TWS_PROMOTED_JOB	Aplica-se às funções de Garantia de Serviço da Carga de Trabalho. Pode ser `YES` ou `No` . Quando o valor é `YES`, significa que a tarefa (uma tarefa crítica ou uma de suas predecessoras) foi promovida.
TWS_TISDIR	O valor de *TWS_TISDIR* definido no ambiente do usuário.
TZ	O fuso horário definido.
UNISON_CPU	O nome desta estação de trabalho.
UNISON_DIR	O valor de *UNISON_DIR* definido no ambiente do usuário.
UNISON_EXEC_PATH	O caminho completo de `.jobmanrc`.
UNISONHOME	O caminho no qual a instância do Tivoli Workload Scheduler foi instalada.
UNISON_HOST	O nome da CPU do host.
UNISON_JOB	O identificador de tarefa absoluto: `worktation#sched_id.job` .
UNISON_JOBNUM	O número do job.
UNISON_MASTER	O nome da estação de trabalho principal.
UNISON_RUN	O número de execução da produção atual do Tivoli Workload Scheduler.
UNISON_SCHED	O nome do fluxo de tarefas.
UNISON_SCHED_DATE	A data de produção (aammdd) do Tivoli Workload Scheduler relatada no cabeçalho do arquivo `Symphony`.
UNISON_SCHED_ID	O *jobstreamID* do fluxo de tarefas que contém a tarefa no processo.
UNISON_SCHED_IA	O *StartTime* do fluxo de tarefas que contém a tarefa no processo.
UNISON_SCHED_EPOCH	A data de produção do Tivoli Workload Scheduler expressa no formato de época.
UNISON_SHELL	O shell de login do usuário executando a tarefa.
UNISON_STDLIST	O nome do caminho do arquivo de lista padrão do job.
UNISON_SYM	O número do registro `Symphony` da tarefa ativada.

Customizando Formatação de Dados no stdlist

É possível utilizar uma variável de ambiente denominada ***UNISON_DATE_FORMAT*** para especificar o formato de data que é utilizada para a data no cabeçalho e no rodapé do arquivo `stdlist`. Essa variável pode ser configurada em estações de trabalho UNIX e Windows e deve ser configurado antes de iniciar processos do Tivoli Workload Scheduler na estação de trabalho para tornar-se efetiva. Para configurar essa variável, siga estas etapas:

Nas estações de trabalho UNIX

1. Inclua a instrução para exportar a variável *UNISON_DATE_FORMAT* no arquivo `.profile` raiz.
2. Execute o arquivo `.profile`.
3. Execute **conman shutdown** e, em seguida, **./StartUp.sh**.

Nas estações de trabalho Windows

1. Em Propriedades do Sistema, configure *UNISON_DATE_FORMAT* na Varável do Sistema.
2. Execute **conman shutdown** e, em seguida, **StartUp**.

Esses são alguns exemplos das configurações utilizadas para exibir o formato de ano no campo de data no cabeçalho e no rodapé do arquivo `stdlist`. A configuração:

```
UNISON_DATE_FORMAT = "%a %x %X %Z %Y"
```

a saída produzida terá o seguinte formato:

```
Fri 15/10/04 11:05:24 AM GMT 2004
```

A configuração:

```
UNISON_DATE_FORMAT = "%a %x %X %Z"
```

a saída produzida terá o seguinte formato:

```
Fri 15/10/04 11:05:24 AM GMT
```

Defina essa variável localmente em cada estação de trabalho para a qual você deseja exibir o formato de ano de 4 dígitos. Se omitido, é utilizado o formato padrão de 2 dígitos.

Customizando o Processamento da Tarefa em uma Estação de Trabalho UNIX - jobmanrc

Um gabarito de script de configuração padrão chamado*TWS_home*/config/**jobmanrc** é fornecido com o Tivoli Workload Scheduler. Ele é instalado automaticamente como *TWS_home*/jobmanrc. Esse script pode ser utilizado pelo administrador do sistema para configurar o ambiente necessário antes de cada tarefa ser executada. Para alterar o script, faça as modificações na cópia de job (*TWS_home*/jobmanrc), deixando o arquivo de gabarito inalterado. O arquivo contém variáveis que podem ser configuradas e comentários para ajudá-lo a entender a metodologia. A Tabela 10 descreve as variáveis de jobmanrc.

Tabela 10. Variáveis Definidas por Padrão no Arquivo jobmanrc

Nome da Variável	Valor
UNISON_JCL	O nome do caminho do arquivo de script do job.
UNISON_STDLIST	O nome do caminho do arquivo de lista padrão do job.
UNISON_EXIT	**yes** | **no** Se definida como **yes**, o job será finalizado imediatamente se algum comando retornar um código de saída diferente de zero. Se definida como **no**, o job continuará a ser executado se um comando retornar um código de saída diferente de zero. Qualquer outra definição será interpretada como **no**.
LOCAL_RC_OK	**yes** | **no** Se configurada como **yes**, o script de configuração local do usuário será executado (se existir), transmitindo*$UNISON_JCL* como o primeiro argumento. Essa opção pode ser permitida ou negada ao usuário. Consulte “Customizando o Processamento de Tarefas para um Usuário nas Estações de Trabalho UNIX - .jobmanrc” na página 49 para obter informações adicionais. Se configurada como **no**, a presença de um script de configuração local será ignorada e *$UNISON_JCL*será executado. Qualquer outra definição será interpretada como **no**.
MAIL_ON_ABEND	**yes** | **no** Para os sistemas operacionais UNIX: Se configurada como **yes**, será enviada uma mensagem para a caixa postal do usuário de login se a tarefa for finalizada com um código de saída diferente de zero. Isso também pode ser definido para um ou mais nomes de usuário, separados por espaços, para que uma mensagem seja enviada para cada um deles. Por exemplo, "root mis sam mary". Se definida como **no**, nenhuma mensagem será enviada se o job abortar. As mensagens de aborto possuem o seguinte formato: *cpu#sched.job* *jcl-file* falhou com *exit-code* Please review *standard-list-filename* É possível alterar o texto da mensagem ou traduzir a mensagem para um outro idioma. Para obter uma explicação de como fazer isso, consulte “Customizando a Seção MAIL_ON_ABEND de jobmanrc” na página 49.

Tabela 10. Variáveis Definidas por Padrão no Arquivo jobmanrc (continuação)

Nome da Variável	Valor
SHELL_TYPE	**standard** | **user** | **script** Se definida como **standard**, a primeira linha do arquivo JCL será lida para determinar qual shell será utilizado para executar o job. Se a primeira linha não iniciar com **#!**, `/bin/sh` será utilizado para executar o script de configuração local ou *$UNISON_JCL*. Os comandos são ecoados para o arquivo de lista padrão do job. Se definida como user, o script de configuração local ou *$UNISON_JCL* será executado pelo shell de login do usuário (*$UNISON_SHELL*). Os comandos são ecoados para o arquivo de lista padrão do job. Se configurada como **script**, o script de configuração local ou *$UNISON_JCL* será executado diretamente e os comandos não serão ecoados, a menos que o script de configuração local ou *$UNISON_JCL* contenha um comando **set -x**. Qualquer outra definição será interpretada como **standard**.
USE_EXEC	**yes** | **no** Se definida como **yes**, o job, ou o script de configuração local do usuário será executado utilizando o comando **exec** eliminando, assim, um processo extra. Essa opção será substituída se *MAIL_ON_ABEND* também estiver configurada como **yes**. Qualquer outra definição será interpretada como **no** e, nesse caso, o job ou script de configuração local será executado por outro processo de shell.

Customizando a Seção MAIL_ON_ABEND de jobmanrc

É possível modificar o texto utilizado na mensagem enviada para os usuários especificados no campo *MAIL_ON_ABEND* do arquivo de configuração *TWS_home*/jobmanrc acessando esse arquivo e alterando o texto nas partes realçadas em negrito:

```
# Mail a message to user or to root if the job fails.

if [ "$MAIL_ON_ABEND" = "YES" ]
then
  if [ $UNISON_RETURN -ne 0 ]
  then
    mail $LOGNAME <<-!
        $UNISON_JOB
        \'$UNISON_JCL\' failed with $UNISON_RETURN
        Please review $UNISON_STDLIST
!
  fi
elif [ "$MAIL_ON_ABEND" = "ROOT" ]
then
  if [ $UNISON_RETURN -ne 0 ]
  then
    mail root <<-!
        $UNISON_JOB
        \'$UNISON_JCL\' failed with $UNISON_RETURN
        Please review $UNISON_STDLIST
!
  fi
elif [ "$MAIL_ON_ABEND" != "NO" ]
then
  if [ $UNISON_RETURN -ne 0 ]
  then
    mail $MAIL_ON_ABEND <<-!
        $UNISON_JOB
        \'$UNISON_JCL\' failed with $UNISON_RETURN
        Please review $UNISON_STDLIST
!
  fi
fi
```

Customizando o Processamento de Tarefas para um Usuário nas Estações de Trabalho UNIX - .jobmanrc

Nas estações de trabalho UNIX, o script de configuração local **.jobmanrc** permite que os usuários estabeleçam um ambiente necessário ao processar suas próprias tarefas. Diferentemente do script **jobmanrc**, o script **.jobmanrc** pode ser personalizado para executar ações diferentes para usuários diferentes. Cada usuário definido como *tws_user* pode customizar no diretório inicial o script **.jobmanrc** para executar ações de pré e pós-processamento. O script **.jobmanrc** é uma etapa extra que ocorre antes que o job seja realmente ativado.

O script **.jobmanrc** é executado somente sob as seguintes condições:

- O script de configuração padrão `jobmanrc` é instalado e a variável de ambiente ***LOCAL_RC_OK*** é configurada para **yes** (consulte a Tabela 10 na página 48).
- Se o arquivo *TWS_home*`/localrc.allow` existir, o nome do usuário deverá aparecer no arquivo. Se o arquivo *TWS_home*`/localrc.allow` não existir, o nome do usuário não deve aparecer no arquivo *TWS_home*`/localrc.deny`. Se nenhum desses arquivos existir, o usuário terá permissão para utilizar um script de configuração local.
- O script de configuração local é instalado no diretório home do usuário (*USER_home*`/.jobmanrc`) e possui permissão de execução.

Os jobs não são executados automaticamente; o comando ou o script devem ser ativados de dentro do **.jobmanrc**. Dependendo do tipo de atividade de processo que você deseja executar, o comando ou o script é ativado de forma diferente. Siga estas regras gerais ao ativar scripts a partir do **.jobmanrc**:

- Utilize **eval** se quiser ativar um comando.
- Utilize **eval** ou **exec** se quiser ativar um script que não precise de atividades pós-processamento.
- Utilize **eval** se quiser ativar um script que precise de atividades pós-processamento.

Se você planeja utilizar um script de configuração local, ele deverá, no mínimo, executar o arquivo de script do job (*$UNISON_JCL*). O Tivoli Workload Scheduler fornece um script de configuração padrão, `jobmanrc`, que executa seu script de configuração local da seguinte forma:

```
$EXECIT $USE_SHELL $USER_home/.jobmanrc "$UNISON_JCL" $IS_COMMAND
```

em que:

- O valor de *USE_SHELL* é definido para o valor da variável `jobmanrc` ***SHELL_TYPE*** (consulte a Tabela 10 na página 48).
- *IS_COMMAND* é configurado para **yes** se a tarefa tiver sido planejada ou enviada na produção utilizando **submit docommand**.
- *EXECIT* será configurado como **exec** se a variável ***USE_EXEC*** estiver configurada como **yes** (consulte a Tabela 10 na página 48), caso contrário ele será nulo.

Todas as variáveis exportadas no **jobmanrc** estão disponíveis no shell **.jobmanrc**; entretanto, as variáveis que estão definidas, mas não exportadas, não estão disponíveis.

O seguinte exemplo mostra como executar o arquivo de script ou o comando de um job, no script de configuração local:

```
#!/bin/ksh
PATH=TWS_home:TWS_home/bin:$PATH
export PATH
/bin/sh -c "$UNISON_JCL"
```

A seguir, o exemplo de um `.jobmanrc` que realiza o processamento com base no código de saída do job do usuário:

```
#!/bin/sh
#
PATH=TWS_home:TWS_home/bin:$PATH
export PATH
/bin/sh -c "$UNISON_JCL"
#or use eval "$UNISON_JCL" and the quotes are required
RETVAL=$?
```

```
if [ $RETVAL -eq 1 ]
then
 echo "Código de saída 1 - Erro Não Fatal"
 exit 0
elif [ $RETVAL -gt 1 -a $RETVAL -lt 100 ]
then
 conman "tellop Este é um erro de banco de dados - pagine o dba"
elif [ $RETVAL -ge 100 ]
then
 conman "Tarefa tellop interrompida. Pagine o admin"
fi
```

Customizando o Processamento de Tarefa em uma Estação de Trabalho Windows - jobmanrc.cmd

Um modelo de script de configuração padrão denominado*TWS_home*\config\jobmanrc.cmd é fornecido com o Tivoli Workload Scheduler. Ele é instalado automaticamente como *TWS_home*\jobmanrc.cmd. Você pode utilizar este arquivo de comando para configurar o ambiente necessário antes de cada tarefa ser executada. Para alterar o arquivo, faça suas modificações na cópia de trabalho (*TWS_home*\jobmanrc.cmd), deixando o arquivo de modelo inalterado. O arquivo contém variáveis que podem ser configuradas e comentários para ajudá-lo a entender a metodologia. Tabela 11 descreve as variáveis de jobmanrc.cmd.

Tabela 11. Variáveis Definidas por Padrão no Arquivo jobmanrc.cmd

Nome da Variável	Valor
HOME	O caminho para o diretório *TWS_home*
POSIXHOME	O caminho para o diretório *TWS_home* em um formato de reclamação de POSIX
LOCAL_RC_OK	• Se configurada para **yes**, o script de configuração local do usuário é executado, se existente. • Se configurada para **no**, a presença de um script de configuração local é ignorada. Qualquer outra definição será interpretada como **no**.
MAIL_ON_ABEND	• Se for configurado como **YES**, um e-mail será enviado ao ID de e-mail definido na variável *email_ID*, se a tarefa terminar com erro. • Se for configurado com outro valor diferente de **YES** ou **NO**, um e-mail será enviado ao ID de e-mail especificado nesta variável se a tarefa terminar com erro. • Se configurada para **NO**, nenhuma mensagem é enviada se a tarefa terminar em erro. Para obter detalhes adicionais, consulte "Customizando a Seção MAIL_ON_ABEND de jobmanrc.cmd".

Customizando a Seção MAIL_ON_ABEND de jobmanrc.cmd

É possível modificar o texto usado na mensagem enviada para os usuários especificados no campo ***MAIL_ON_ABEND*** do arquivo de configuração *TWS_home*/jobmanrc.cmd acessando esse arquivo e alterando o texto nas partes destacadas em negrito. Para esclarecer como gerar a mensagem de e-mail, é usado um programa de correio de amostra com o nome bmail.exe.

```
if /I "%MAIL_ON_ABEND%"=="NO" (goto :out) else (goto :mail_on_abend)

:mail_on_abend
REM ******email, task or other action inserted here ******************
if /I "%MAIL_ON_ABEND%"=="YES" (goto :email) else (goto :email_spec)

:email
c:\"Program Files"\utils\bmail.exe -s smtp.yourcompany.com -t %EMAIL_ID%
-f %COMPUTERNAME%@yourcompany.com -h -a "Subject: Job %UNISON_JOB% abended"
-b "Job %UNISON_JOB% Job Number %UNISON_JOBNUM% abended"
goto :out

:email_spec
```

```
REM set > c:\tmp\abended_jobs\%UNISON_JOB%.j%UNISON_JOBNUM%
c:\"Program Files"\utils\bmail.exe -s smtp.yourcompany.com -t %MAIL_ON_ABEND%
-f %COMPUTERNAME%@yourcompany.com -h -a "Subject: Job %UNISON_JOB% abended"
-b "Job %UNISON_JOB% Job Number %UNISON_JOBNUM% abended"
```

Customizando Processamento de Tarefa em uma Estação de Trabalho com o Windows - djobmanrc.cmd

Nas estações de trabalho com o Windows, é possível usar o script de configuração local djobmanrc.cmd para estabelecer um ambiente específico ao processar suas tarefas customizadas. Diferentemente do script jobmanrc.cmd, é possível customizar o script djobmanrc.cmd para executar diferentes ações para diferentes usuários.

As condições a seguir se aplicam:

- O script deve conter todas as variáveis de aplicativos de ambiente ou caminhos necessários para que o Tivoli Workload Scheduler seja ativado corretamente.
- O script deve existir se um ambiente específico do usuário para execução da tarefa for necessário ou se um e-mail deve ser enviado ao usuário de logon da tarefa quando a tarefa Tivoli Workload Scheduler terminar com erro.

Para criar um script djobmanrc.cmd customizado, execute as etapas a seguir:

1. Efetue logon como o usuário que define variáveis de ambiente para ativar tarefas do Tivoli Workload Scheduler.
2. Abra um prompt de comandos do DOS.
3. Digite o comando **set** redirecionando a saída padrão para um arquivo simples denominado`user_env`.
4. Crie um arquivo denominado djobmanrc.cmd no diretório Documentos e Configurações do usuário com o texto padrão a seguir no início:

   ```
   @ECHO OFF
       echo Invoking %USERNAME% DJOBMANRC.CMD V.1
       set USERPROFILE=%USERPROFILE%
       ::Setup User Environment Phase
   ```

5. Edite o arquivo `user_env` criando na etapa 3.
6. Insira o comando **set** em cada linha antes de cada variável de ambiente.
7. Inclua as mudanças na variável PATH no final de djobmanrc.cmd em uma sequência semelhante à seguinte:

   ```
   set PATH=<TWSHOME>;<TWSHOME>\bin;%PATH%
   ```

8. Inclua o texto a seguir no final do arquivo `user_env` e substitua a sequência *ID do e-mail do usuário* com o ID de e-mail do usuário que recebe a notificação por e-mail se a tarefa terminar com erro.

   ```
   set EMAIL_ID=<user email id>
       ::Launch Operation Phase
       %ARGS%
       ::Post Operations Phase
       :out
   ```

9. Inclua o arquivo `user_env` atualizado no final do arquivo djobmanrc.cmd. O arquivo djobmanrc.cmd editado deve ter a aparência do exemplo a seguir:

   ```
   @ECHO OFF
       echo Invoking %USERNAME% DJOBMANRC.CMD V.1
       set USERPROFILE=%USERPROFILE%
       ::Setup User Environment Phase
       set ALLUSERSPROFILE=C:\Documents and Settings\All Users
       set APPDATA=C:\Documents and Settings\petes\Application Data
       set CommonProgramFiles=C:\Program Files\Common Files
       set COMPUTERNAME=PSOTOJ
       set ComSpec=C:\WINDOWS\system32\cmd.exe
       set CURDRIVE=C
       set FP_NO_HOST_CHECK=NOset
       set HOMEDRIVE=c:
       set HOMEPATH=\docs
       set LOGONSERVER=\\PSOTOJ
       set NEWVAR=c:\tmp\tmp\mlist1
       set NUMBER_OF_PROCESSORS=1
       set OPC_CLIENT_ROOT=C:\opc\Client
   ```

```
set OS=Windows_NT
set Path=C:\Program Files\utils;C:\PROGRAM
FILES\THINKPAD\UTILITIES;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\Program
Files\IBM\Infoprint Select;C:\Utilities;C:\Notes;C:\Program Files\IBM\Trace Facility\;C:\Program
Files\IBM\Personal Communications\;C:\Program Files\XLView\;C:\lotus\compnent\;C:\WINDOWS\Downloaded
Program Files;C:\Program Files\Symantec\pcAnywhere\;"C:\Program Files\Symantec\Norton Ghost
2003\";C:\Infoprint;
set PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
set PCOMM_Root=C:\Program Files\IBM\Personal Communications\
set PDBASE=C:\Program Files\IBM\Infoprint Select
set PDHOST=
set PD_SOCKET=6874
set PROCESSOR_ARCHITECTURE=x86
set PROCESSOR_IDENTIFIER=x86 Family 6 Model 9 Stepping 5, GenuineIntel
set PROCESSOR_LEVEL=6
set PROCESSOR_REVISION=0905
set ProgramFiles=C:\Program Files
set PROMPT=$P$G
set SESSIONNAME=Console
set SystemDrive=C:
set SystemRoot=C:\WINDOWS
set TEMP=C:\DOCUME~1\petes\LOCALS~1\Temp
set TMP=C:\DOCUME~1\petes\LOCALS~1\Temp
set tvdebugflags=0x260
set tvlogsessioncount=5000
set TWS4APPS_JDKHOME=c:\win32app\TWS\pete\methods\_tools
set USERDOMAIN=PSOTOJ
set USERNAME=petes
set USERPROFILE=C:\Documents and Settings\petes
set windir=C:\WINDOWSPATH=c:\win32app\tws\twsuser:c:\win32app\tws\twsuser\bin:%PATH%
set PATH=c:\win32app\TWS\twsuser;c:\win32app\TWS\twsuser\bin;%PATH%
set EMAIL_ID=johndoe@yourcompany.com
::Launch Operation Phase
%ARGS%
::Post Operations Phase
:out
```

A Launch Operations Phase no script é onde o script, binário ou comando definido para a tarefa é concluído. O texto "%ARGS%" é obrigatório.

A Post Operations Phase no script é onde um código de saída da tarefa pode ser reajustado do estado ABEND para o SUCC, alterando um código de saída diferente de zero para um código de saída zero. Alguns aplicativos podem ter códigos de saída que podem ser avisos. O Tivoli Workload Scheduler avalia os códigos de saída como zero ou diferente de zero. Códigos de saída zero indicam uma tarefa no estado "SUCC". Todos os outros códigos indicam uma tarefa no estado ABEND. Especifique que códigos de saída diferentes de zero podem ser ajustados, se necessário. O exemplo a seguir mostra o que pode ser incluído na Post Operations Phase. O exemplo recupera o código de saída da tarefa definida para determinar como tratar itbased nas instruções `If`:

```
set EMAIL_ID=johndoe@yourcompany.com
    ::Launch Operation Phase
    %ARGS%
    ::Post Operations Phase
    set RETVAL=%ERRORLEVEL%
    if "%RETVAL%"=="0" goto out
    if "%RETVAL%"=="1" set RETVAL=0
    if "%RETVAL%"=="6" set RETVAL=0
    :out
    exit %RETVAL%
```

Configurando Opções para Utilização das Interfaces com o Usuário

Para usar oDynamic Workload Console, os parâmetros de conexão são fornecidos no console e salvos como parte de sua configuração.

Para usar oTivoli Workload Scheduler cliente da linha de comandos, é necessário fornecer as informações de configuração a seguir (chamadas de ***connection_parameters***) para conectar ogerenciador de domínio principal por meio de HTTP/HTTPS usando a infraestrutura do WebSphere Application Server:

hostname
O nome do host do gerenciador de domínio principal.

número de porta
O número da porta usado ao estabelecer a conexão com o gerenciador de domínio principal.

nome de usuário, senha
As credenciais, nome de usuário e senha, do *TWS_user*.

proxy hostname
O nome do host do proxy utilizado na conexão com o protocolo HTTP.

proxy port number
O número da porta do proxy utilizado na conexão com o protocolo HTTP.

protocol
O protocolo utilizado durante a comunicação. Pode ser HTTP com autenticação básica ou HTTPS com autenticação de certificado.

timeout
O tempo limite que indica o tempo máximo que o programa da interface com o usuário que está conectando pode esperar pela resposta do gerenciador de domínio principal antes de considerar a solicitação de comunicação como com falha.

default workstation
O nome da estação de trabalho do gerenciador de domínio principal à qual deseja conectar.

SSL parameters
Se tiver configurado sua rede para usar SSL para se comunicar entre as interfaces e ogerenciador de domínio principal, também é necessário fornecer o conjunto apropriado de parâmetros SSL (que depende de como seu SSL é configurado).

No caso do cliente da linha de comandos instalado no gerenciador de domínio principal, essa configuração é executada automaticamente na instalação.

Para o cliente da linha de comandos instalado em outras estações de trabalho, essas informações podem ser fornecidas armazenando-as em arquivos de propriedades nessas estações de trabalho ou fornecendo as informações como parte da sequência de caracteres de comando dos comandos usados.

Os arquivos de propriedades referidos são os arquivos **localopts** e **useropts**:

localopts
Isso contém um conjunto de parâmetros aplicável à estação de trabalho local para uma instância específica do produto instalado.

useropts
Isso contém um subconjunto dos parâmetros localopts que têm valores customizados para um usuário específico. O caminho desse arquivo está dentro do diretório inicial do usuário, que mantém a privacidade dessas informações.

Como oTivoli Workload Scheduler suporta diversas instâncias de produtos instaladas na mesma máquina, pode haver mais e uma instância do arquivo `useropts` de cada usuário. A possibilidade de ter mais arquivos `useropts`, cada um com um nome diferente, fornece a capacidade de especificar diferentes conjuntos de configurações de conexão para usuários definidos em mais de uma instância do produto instalada na mesma máquina.

No arquivo `localopts` de cada instância do produto instalado, a opção denominada ***useropts*** identifica o nome do arquivo `useropts` que precisa ser acessado para conectar a essa instância de instalação.

Isso significa que, se duas instâncias do Tivoli Workload Scheduler forem instaladas em uma máquina e um usuário do sistema denominado`operator` for definido como usuário em ambas as instâncias, então no arquivo `localopts` da primeira instância, a opção local *useropts* = `useropts1` identifica o arquivo `useropts1` que contém as configurações de parâmetros de conexão que o usuário `operator` precisa para usar para conectar a essa instância do Tivoli Workload Scheduler. Por outro lado, no arquivo `localopts` da segunda instância do Tivoli Workload Scheduler, a opção local *useropts* = `useropts2` identifica o arquivo `useropts2` que contém as configurações de parâmetros de conexão que o usuário `operator` precisa para usar para conectar a essa instância do Tivoli Workload Scheduler.

Detalhes completos sobre como configurar esse acesso são fornecidos no *Tivoli Workload Scheduler: Administration Guide*, no tópico intitulado "Configurando Autenticação de Acesso ao Cliente de Linha de Comandos"

Capítulo 4. Gerenciando o Ciclo de Produção

A parte principal de uma solução de gerenciamento e planejamento de tarefas é a criação e o gerenciamento do *plano de produção*. O plano de produção é a lista de afazeres que contém as ações a serem executadas em um intervalo de tempo determinado nas estações de trabalho da rede de planejamento utilizando os recursos disponíveis e preservando os relacionamentos e restrições definidas.

Este capítulo descreve como o Tivoli Workload Scheduler gerencia planos.

Este capítulo está dividido nas seções a seguir:

- "Conceitos Básicos de Gerenciamento de Planos"
- "Customizando o Gerenciamento de Planos Utilizando Opções Globais" na página 75
- "Criando e estendendo o plano de produção" na página 79
- "Linha de Comandos Planman" na página 82
- "Iniciando o Processamento do Plano de Produção" na página 100
- "Automatizando o processamento do plano de produção" na página 101

Conceitos Básicos de Gerenciamento de Planos

O *plano de produção* contém informações sobre as tarefas a serem executadas, em qual agente tolerante a falhas e quais dependências devem ser atendidas antes de cada tarefa ser ativada. O Tivoli Workload Scheduler cria o plano de produção a partir dos dados de modelagem do banco de dados e de um plano intermediário chamado *plano de pré-produção*. O plano de pré-produção é criado e gerenciado automaticamente pelo produto. Para evitar problemas, o banco de dados é bloqueado durante a geração do plano e é desbloqueado quando a geração é concluída ou se ocorrer uma condição de erro. O plano de pré-produção é utilizado para identificar antecipadamente as instâncias de fluxo de tarefas e as dependências de fluxo de tarefas follows externas envolvidas em um janela de tempo especificada.

Você utiliza o script **JnextPlan** no gerenciador de domínio principal para gerar o plano de produção e distribuí-lo pela rede do Tivoli Workload Scheduler. Para obter informações adicionais sobre o script **JnextPlan**, consulte "Criando e estendendo o plano de produção" na página 79.

Para gerar e iniciar um novo plano de produção, o Tivoli Workload Scheduler executa as seguintes etapas:

1. Atualiza o plano de pré-produção com os objetos definidos no banco de dados que foram incluídos ou atualizados desde a última vez que o plano foi criado ou estendido.
2. Recupera do plano de pré-produção as informações sobre o fluxo de tarefas a ser executado no período de tempo especificado e salva o mesmo em um plano de produção intermediário.
3. Inclui no novo plano de produção os fluxos de tarefas incompletos do plano de produção anterior.

© Copyright IBM Corp. 1999, 2014

4. Cria o novo plano de produção e armazena o mesmo em um arquivo denominado Symphony. Os dados do plano também são replicados no banco de dados.
5. Distribui uma cópia do arquivo Symphony para as estações de trabalho envolvidas no processamento do novo plano de produto.
6. Registra todas as estatísticas do plano de produção anterior em um archive
7. Atualiza os estados de fluxo de tarefas no plano pré-produção.

A cópia do novo arquivo Symphony gerado é implementada a partir dos agente tolerante a falhas do domínio inicial e dos gerenciadores de domínio dos domínios filhos até todos os domínios subordinados abaixo na árvore.

Cada agente tolerante a falhas e gerenciador de domínio que recebe o novo arquivo Symphony, arquiva o Symphony anterior em Symphony.last no caminho <*TWA_home*>/TWS/, para que uma cópia de backup seja mantida. Isso permite a visualização dos dados do Symphony anterior em caso de alguma atualização de mensagem nos estados da tarefa e do fluxo de tarefas que foram perdidos entre o agente e seu gerenciador de domínio principal.

Cada agente tolerante a falhas que recebe o plano de produção pode continuar o processamento mesmo se a conexão de rede com seu gerenciador de domínio ficar inativa.

Em cada agente tolerante a falhas de destino, os processos do Tivoli Workload Scheduler executam as seguintes ações para gerenciar o processamento de tarefa:

1. Acessar a cópia do arquivo Symphony e ler as instruções sobre quais tarefas executar.
2. Fazer chamadas ao sistema operacional para ativar tarefas conforme requeridas.
3. Atualizar sua cópia do arquivo Symphony com os resultados do processamento de tarefa e enviar a notificação de volta ao gerenciador de domínio principal e a todos os agente tolerante a falhass com status completo. A cópia original do arquivo Symphony armazenada no gerenciador de domínio principal e as cópias armazenadas nos gerenciador de domínio principals de backup, se definidas, são atualizadas conforme necessário.

Isto significa que, durante o processamento de tarefa, cada agente tolerante a falhas possui sua própria cópia do arquivo Symphony atualizado com as informações sobre as tarefas que está executando (ou que estão em execução em seu domínio e domínios-filhos se o agente tolerante a falhas tiver o status completo ou um gerenciador de domínio). Além disso, o gerenciador de domínio principal (e o gerenciador de domínio principal de backup, se definido) possui a cópia do arquivo Symphony que contém todas as atualizações provenientes de todos os agente tolerante a falhass. Dessa forma, o arquivo Symphony no gerenciador de domínio principal é mantido atualizado com as tarefas que precisam ser executadas, as que estão em execução e as que foram concluídas.

O processamento que ocorre em cada estação de trabalho envolvida nas atividades do plano de produção atual é descrito mais detalhadamente em “Processos da Estação de Trabalho do Tivoli Workload Scheduler” na página 31.

Nota: Enquanto o plano de produção atual está em andamento, as mudanças feitas no plano usando **conman** não afetam as definições no banco de dados, mas a réplica dos dados do plano no banco de dados é atualizada com as mudanças. Alterações feitas nos objetos no banco de dados não afetam o plano até o plano de

produção ser estendido ou criado novamente utilizando o script **JnextPlan** ou a interface de linha de comandos **planman**. As atualizações dos objetos no banco de dados não afetam instâncias desses objetos que já se encontram no plano de produção.

Plano de Pré-Produção

O plano de pré-produção é utilizado para identificar antecipadamente as instâncias do fluxo de tarefas e as dependências do fluxo de tarefas envolvidas em um período de tempo especificado.

Isso aprimora o desempenho ao gerar o plano de produção preparando antecipadamente um planejamento de alto nível da carga de trabalho de produção antecipada.

O plano de pré-produção contém:

- As instâncias do fluxo de tarefas a serem executadas durante o intervalo de tempo coberto.
- As dependências follows externas que existem entre os fluxos de tarefa e as tarefas incluídas nos fluxos de tarefa diferentes.

Uma tarefa ou fluxo de tarefas que não pode ser iniciado antes de outra tarefa ou fluxo de tarefas externo específico ser concluído com êxito é denominado*sucessor*. Uma tarefa ou fluxo de tarefas externo que deve ser concluído com êxito antes da tarefa ou fluxo de tarefas sucessor poder ser iniciado é denominado*predecessor*.

O Tivoli Workload Scheduler gera, expande e atualiza automaticamente, se necessário, o plano de pré-produção, executando as seguintes etapas:

- Remove as instâncias do fluxo de tarefas nos estados COMPLETE e CANCEL.
- Seleciona todos os fluxos de tarefas planejados após o final do plano de produção atual e gera suas instâncias.
- Resolve todas as dependências de tarefa e de fluxo de tarefas, incluindo dependências follows externas de acordo com os critérios de correspondência definidos.

Para evitar conflitos, o banco de dados é bloqueado durante a geração do plano de pré-produção e desbloqueado quando a geração é concluída ou se ocorrer uma condição de erro.

Nesse estágio, somente os fluxos de tarefas com a hora na qual estão planejados para iniciar e suas dependências são destacados. Todas as informações restantes sobre os fluxos de tarefas e os outros objetos de planejamento (calendários, avisos, domínios, estações de trabalho, recursos, arquivos e usuários) que estarão envolvidos no plano de produção para esse período de tempo não estão incluídas, mas são recuperadas do banco de dados assim que o plano de produção for gerado.

Quando o plano de produção for estendido, as instâncias de fluxo de tarefa antigas são removidas automaticamente. Os critérios utilizados na remoção dessas instâncias levam em consideração estas informações:

- A primeira instância do fluxo de tarefas que não está no estado COMPLETE na hora que o novo plano é gerado (FNCJSI). Essa instância do fluxo de tarefas pode ser uma instância planejada, que é uma instância incluída no plano quando o plano de produção é gerado, e uma instância do fluxo de tarefas, enviada da linha de comandos durante a produção utilizando o comando **conman sbs**.

- O período de tempo entre a hora que FNCJSI está planejado para iniciar e o horário de encerramento do plano de produção antigo.

Supondo que **T** seja esse período de tempo, o algoritmo utilizado para calcular quais instâncias do fluxo de tarefas são removidas do plano de pré-produção é o seguinte:

if T < 7
: Todas as instâncias do fluxo de tarefas mais antigas do que 7 dias do horário de início do novo plano de produção são removidas do plano de pré-produção; todas as instâncias do fluxo de tarefas mais próxima do que 7 dias do horário de início do novo plano de produção são mantidas independentemente de seus estados.

if T > 7
: Todas as instâncias do fluxo de tarefas mais antigas do que FNCJSI são removidas do plano de pré-produção; todas as instâncias do fluxo de tarefas mais recentes do que FNCJSI são mantidas.

Esse algoritmo é utilizado para assegurar que o tamanho do plano de pré-produção não aumente de forma contínua e, ao mesmo tempo, para assegurar que nenhuma instância do fluxo de tarefas que seja um predecessor em potencial de um novo fluxo de tarefas incluído no novo plano de pré-produção seja excluída.

Para obter informações adicionais sobre como é possível abrir o plano de pré-produção no modo de visualização a partir do Dynamic Workload Console, consulte o Guia do Usuário do Console do Dynamic Workload, seção sobre Visualizar plano de pré-produção.

Nota: Na terminologia do Tivoli Workload Scheduler para z/OS, o conceito que corresponde ao plano de pré-produção é LTP (*Long Term Plan*).

Identificando Instâncias do Fluxo de Tarefas no Plano

Em versões anteriores à 8.3, o plano tinha uma duração fixa de um dia. Desde a versão 8.3, o plano pode cobrir um período que dura vários dias ou menos de um dia. Esta alteração incluiu a possibilidade de ter no mesmo plano mais de uma instância do mesmo fluxo de tarefas com o mesmo nome e também a necessidade para definir uma nova convenção para identificar exclusivamente cada instância do fluxo de tarefas no plano. Cada instância do fluxo de tarefas é identificada no plano pelos valores a seguir:

estação de trabalho
: Especifica o nome da estação de trabalho na qual o fluxo de tarefas está planejado para execução.

jobstreamname
: Corresponde ao nome do fluxo de tarefas utilizado nas versões anteriores do Tivoli Workload Scheduler.

scheddateandtime
: Representa quando a instância do fluxo de tarefas está planejada para iniciar no plano de pré-produção. Corresponde ao dia especificado no ciclo de execução configurado na definição do fluxo de tarefas por uma cláusula **on** e a hora configurada na definição do fluxo de tarefas por uma palavra-chave **at** ou **schedtime**. Se configurada, a palavra-chave **schedtime** é utilizada somente para ordenar cronologicamente as instâncias do fluxo de tarefas no plano de pré-produção, enquanto que, se configurada, a palavra-chave **at** também representa uma dependência para o fluxo de

tarefas. Para obter informações adicionais sobre essas palavras-chave, consulte “em” na página 244, “às” na página 220 e “schedtime” na página 253.

Juntamente com esses dois valores que podem ser configurados na definição do fluxo de tarefas, o Tivoli Workload Scheduler gera e designa um identificador alfanumérico exclusivo para cada instância do fluxo de tarefas, o ***jobstream_id***, para seu processamento interno. Para obter informações adicionais sobre o formato do *jobstream_id*, consulte “showjobs” na página 411.

É possível utilizar qualquer um dos dois tipos de identificadores, *workstation#jobstreamname* e *scheddateandtime*, em vez de *workstation#jobstream_id*, para identificar de forma exclusiva uma instância do fluxo de tarefas ao gerenciar fluxos de tarefas no plano, utilizando o programa de linha de comandos **conman**. A convenção padrão utilizada para identificar uma instância do fluxo de tarefas, neste guia e nas interfaces de linha de comandos do produto, é a que utiliza *workstation#jobstreamname* e *scheddateandtime*. Para obter informações adicionais sobre como especificar uma instância de fluxo de tarefa em um comando utilizando **conman**, consulte “Selecionando Fluxos de Tarefas em Comandos” na página 353.

Gerenciando Dependências de Acompanhamentos Externos para Tarefas e Fluxos de Tarefas

Durante a criação do plano de pré-produção, todas as dependências follows externas para fluxos de tarefas e tarefas são resolvidas utilizando-se quatro *critérios de correspondência* diferentes possíveis:

Mesmo dia
Considerando as instâncias da tarefa ou do fluxo de tarefas planejadas para execução no mesmo dia. Neste caso, configure a cláusula **follows**...**sameday** na definição de objeto. Figura 6 mostra um fluxo de tarefa denominado `Js1` que possui uma dependência follows externa na instância do fluxo de tarefa `Js2` que está planejada para iniciar no mesmo dia.

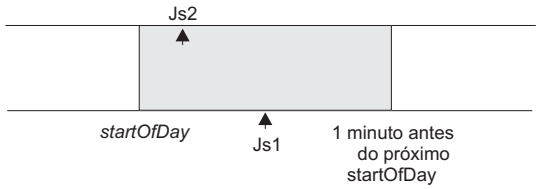


Figura 6. Critérios de Correspondência do Mesmo Dia

Abaixo há um exemplo de como definir os fluxos de tarefa envolvidos.

schedule Js2
on everyday
at 0700
:job2
end

schedule Js1
on everyday
at 1000
follows wk1#Js2 sameday
:job1
end

O fluxo de tarefa `Js1` não é ativado até que a instância do fluxo de tarefas de `Js2` na estação de trabalho `wk1` seja concluída com êxito.

Precedente mais próximo

Usando a tarefa mais próxima ou a instância do fluxo de tarefas (anteriormente ou mesmo horário). A instância da tarefa ou do fluxo de tarefas que o Tivoli Workload Scheduler usa para resolver a dependência é a mais próxima em tempo que precede a instância que inclui a dependência. Neste caso, você configura a cláusula **follows ... previous** na definição de objeto. Figura 7 mostra um fluxo de tarefa denominado `Js1` que possui uma dependência follows externa na instância mais antiga mais próxima do fluxo de tarefa `Js2`. O período de tempo em que o predecessor é procurado é esmaecido na figura.

Figura 7. Critérios de Correspondência Anteriores Mais Próximos

Abaixo há um exemplo de como definir os fluxos de tarefa envolvidos.

schedule Js2	**schedule Js1**
em Th	em Fr
at 0700	at 1000
:job2	follows wk1#Js2 previous
end	:job1
	end

O fluxo de tarefa `Js1` não é ativado até que a instância do fluxo de tarefas precedente mais próxima de `Js2` na estação de trabalho `wk1` seja concluída com êxito.

Dentro de um intervalo relativo

Considerando as instâncias da tarefa ou fluxo de tarefa definidas em um intervalo com um deslocamento relativo ao horário de início da tarefa ou fluxo de tarefa dependente. Por exemplo, de 25 horas antes do horário de início do fluxo de tarefas dependente até 5 horas após o horário de início do fluxo de tarefas dependente. Neste caso, configure a cláusula **follows ... relative from ... to ...** na definição de objeto. A Figura 8 mostra um fluxo de tarefas denominado `Js1` que possui uma dependência follows externa na instância do fluxo de tarefas de `Js2` que inicia com um deslocamento de 2 horas em relação a `Js1`. A instância da tarefa ou do fluxo de tarefas que o Tivoli Workload Scheduler considera para resolver a dependência é a mais próxima dentro do intervalo de tempo relativo escolhido.

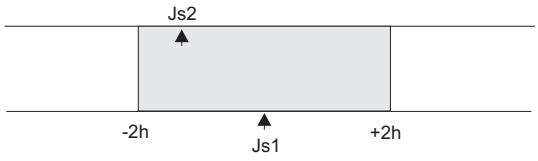


Figura 8. Nos Critérios de Correspondência de um Intervalo Relativo

Abaixo há um exemplo de como definir os fluxos de tarefa envolvidos.

schedule Js2	**schedule Js1**
on everyday	on everyday
às 0900	at 1000

schedule Js2	**schedule Js1**
:job2	follows wk1#Js2 relative from 0200 to 0200
end	:job1
	end

O fluxo de tarefa Js1 não é ativado até que a instância do fluxo de tarefas de Js2 na estação de trabalho wk1 que é executada no intervalo de tempo 08h00 a 12h00 seja concluída com êxito.

Dentro de um intervalo absoluto

Utilizando apenas as instâncias de tarefa ou do fluxo de tarefas definidas em um intervalo. Por exemplo, de hoje às 6h00 até o dia depois de amanhã às 5h59. Neste caso, você configura a cláusula **follows ... from ... to ...** na definição de objeto. A Figura 9 mostra um fluxo de tarefas chamado Js1 que possui uma dependência follows externa na instância do fluxo de tarefas Js2 que está posicionada no plano de pré-produção entre 7h e 9h. A instância da tarefa ou do fluxo de tarefas que o Tivoli Workload Scheduler considera para resolver a dependência é a mais próxima dentro de um intervalo de tempo absoluto escolhido. O intervalo de tempo especifica a hora do dia na qual o intervalo começa e termina, no mesmo dia que a instância que inclui a dependência ou em um dia definido como relativo para esse dia.

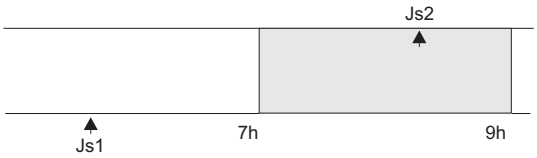


Figura 9. Nos Critérios de Correspondência de um Intervalo Absoluto

Abaixo há um exemplo de como definir os fluxos de tarefa envolvidos.

schedule Js1	**schedule Js2**
on everyday	on everyday
at 0800	at 1000
:job2	follows wk1#Js1 from 0700 to 0900
end	:job1
	end

O fluxo de tarefa Js1 não é ativado até que a instância do fluxo de tarefas de Js2 na estação de trabalho wk1 que é executada no intervalo de tempo das 7h às 9h no mesmo dia seja concluída com êxito.

Independentemente de quais critérios correspondentes são utilizados, se várias instâncias de fluxos de tarefas do predecessor em potencial existirem no intervalo de tempo especificado, a regra utilizada pelo produto para identificar a instância do predecessor correta será a seguinte:

1. Tivoli Workload Scheduler procura a instância mais próxima que precede o horário de início da tarefa ou do fluxo de tarefas dependente. Se uma instância desse tipo existir, esta será a instância do predecessor.
2. Se não houver instância precedente, o Tivoli Workload Scheduler considerará a instância do predecessor correta como a instância mais próxima iniciada após o horário de início da tarefa ou do fluxo de tarefas dependente.

Esse comportamento aplica-se às dependências follows externas entre fluxos de tarefas. Para dependências follows externas de um fluxo de tarefas ou tarefa a partir de outra tarefa, os critérios são correspondidos pela consideração do horário de início do fluxo de tarefas que faz o hosting da tarefa predecessora em vez do horário de início da própria tarefa predecessora. A Figura 10 mostra em negrito as instâncias `job1` das quais a tarefa ou o fluxo de tarefas sucessor depende.

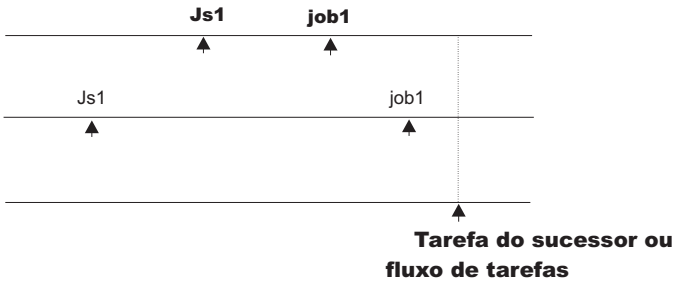


Figura 10. Tarefa Predecessora Anterior Mais Próxima

As dependências follows externas são identificadas entre tarefas e fluxos de tarefas armazenados no banco de dados cujas instâncias são incluídas no plano de pré-produção quando o plano de pré-produção é criado ou estendido automaticamente. As instâncias da tarefa e do fluxo de tarefas enviadas na produção a partir da linha de comandos **conman** são gravadas no plano de pré-produção mas elas não são utilizadas para recalcular os predecessores de dependências follows externas já resolvidas no plano de pré-produção.

O planejador classifica dependências follows como *internas* quando são especificadas apenas por seu nome da tarefa dentro do fluxo de tarefas. Ele classifica-as como *externas* quando são especificadas no formato *jobStreamName.workstationName.jobName*.

Quando um fluxo de tarefas inclui uma tarefa com uma dependência de sequência que compartilha o mesmo nome de fluxo de tarefas (por exemplo, o fluxo de tarefas `schedA` inclui uma tarefa denominada `job6` que tem uma dependência de sequência em `schedA.job2`), a dependência é incluída no plano como uma dependência de sequência *externa*. Desde a Versão 8.3, ao contrário das versões anteriores, porque o planejador usa os critérios de correspondência `mesmo dia` para resolver dependências externas, dependências originadas dessa maneira nunca são incluídas da primeira vez que o objeto é enviado.

Uma tarefa ou um fluxo de tarefas ainda não incluído no plano de produção, mas que pode ser um predecessor em potencial de instâncias de tarefas e de fluxos de tarefas, incluído no plano de produção à medida que o plano de produção é estendido, é chamado de ***predecessor pendente***. Um predecessor pendente é como uma ocorrência simulada criada pelo processo de planejamento para honrar uma dependência que foi resolvida no plano de pré-produção, mas isso não pode ser resolvido no plano de produção, pois o horário de início do predecessor não está dentro do horário de encerramento do plano de produção atual. A Figura 11 na página 65 mostra como um predecessor pendente e seu sucessor são posicionados no plano de pré-produção.

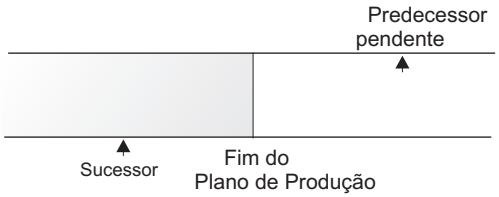


Figura 11. Instância do Predecessor Pendente

A maneira como os predecessores pendentes são gerenciados está estritamente vinculada a se a tarefa ou o fluxo de tarefas sucessor teve ou não o saldo transportado:

- Se o sucessor tiver o saldo transportado quando o plano de produção for estendido, o predecessor é incluído no novo plano de produção e a dependência torna-se atual. Uma tarefa ou um fluxo de tarefas de predecessor pendente é marcado com `[P]` na coluna `Dependências`, na saída dos comandos conman showjobs e conman showschedules.
- Se o sucessor não tiver o saldo transportado adiante quando o plano de produção for estendido, o predecessor é incluído no novo plano de produção, mas a dependência torna-se ***órfã***. Isso pode ocorrer, por exemplo, se, ao estender o plano de produção, o sucessor tiver o saldo transportado e o predecessor pendente não for incluído no plano, pois foi sinalizado como *rascunho* no banco de dados. As dependências órfãs são marcadas com `[O]` na coluna `Dependências`, na saída do comando conman showjobs. Ao lidar com uma dependência órfã, você deve verificar se pode ser liberada e, nesse caso, cancelá-la.

Observe que quando uma rede do Tivoli Workload Scheduler inclui agentes que estão executando versões mais antigas que 8.3 e a opção `enLegacyId` está configurada para `yes` no gerenciador de domínio principal, ter várias instâncias de um fluxo de tarefas como predecessores pendentes produz os erros causados por problemas de identificação no momento do envio.

Exemplos de Resolução de Dependência de Sequência Externa e de Transição de Status

Esta seção inclui exemplos para cada um dos quatro critérios de correspondência descritos nos parágrafos anteriores. Em todos os exemplos, o horário de início do dia (SOD) está configurado para 6h.

Mesmo dia

A instância da tarefa ou do fluxo de tarefas a ser considerada na resolução da dependência é a mais próxima no mesmo dia em que a instância que inclui a dependência está programada para execução. Neste exemplo, dois fluxos de tarefas, `Js1` e `Js2`, têm, cada um, uma tarefa. O fluxo de tarefas `Js1` está planejado para execução todo dia às 8h e nas quintas-feiras também às 7h. `Js1.Job1` é executada às 9h. O fluxo de tarefas `Js2` não tem nenhuma restrição de tempo e esta planejado por padrão no horário de início do dia definido. A `Js2.Job2` está planejada para execução às 15h e tem uma dependência de sequência externa na instância anterior mais próxima do fluxo de tarefas `Js1` em execução no mesmo dia. Os dois fluxos de tarefas são definidos desta maneira:

```
SCHEDULE MY_MASTER#JS1
ON RUNCYCLE RULE1 "FREQ=WEEKLY;BYDAY=TH"
(AT 0700)
ON RUNCYCLE RULE2 "FREQ=DAILY"
(AT 0800)
:
MY_MASTER#JOB1
AT 0900
END
```

```
SCHEDULE MY_MASTER#JS2
ON RUNCYCLE RULE2 "FREQ=DAILY;"
FOLLOWS MY_MASTER#JS1.@ SAMEDAY
:
MY_MASTER#JOB2
AT 1500
END
```

Quando os planejamentos são incluídos no plano, a sequência de gráficos ilustra como a dependência é resolvida:

1. Nas quintas-feiras, a instância de `Js2` planejada às 6h depende da instância de `Js1` planejada para execução às 7h. Em qualquer outro dia da semana, `Js2` tem uma dependência da instância de `Js1` planejada às 8h. A Figura 12 mostra o status dos fluxos de tarefas no plano às 6h (SOD) na quinta-feira:

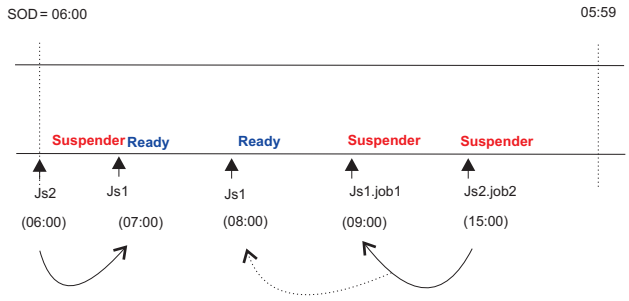


Figura 12. Critérios de Correspondência no Mesmo Dia - Etapa 1: no Início do Dia (SOD) em uma Quinta-feira

2. Às 9h, `Js1.job1` inicia e `Js1` altera o status. `Js2.job2` é retida até seu horário planejado. A Figura 13 mostra o status dos fluxos de tarefas no plano às 9h.

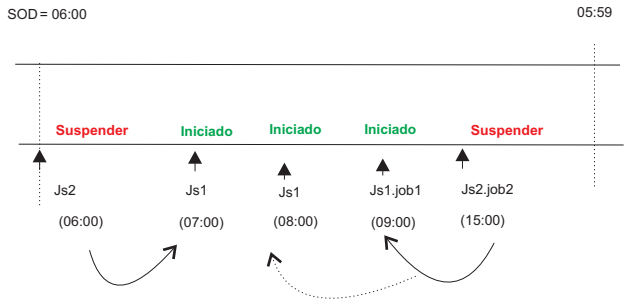


Figura 13. Critérios de Correspondência no Mesmo Dia - Etapa 2: às 9h

3. Nas quintas-feiras às 15h, `Js2` altera para o status pronto e `Js2.job2` inicia. A Figura 14 na página 67 mostra o status dos dois fluxos de tarefas no plano às 15h.

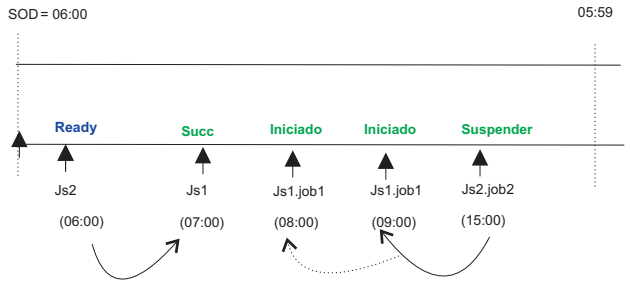


Figura 14. Critérios de Correspondência no Mesmo Dia - Etapa 3: às 15h

Precedente mais próximo

- Neste exemplo, dois fluxos de tarefas, `Js1` e `Js2`, têm, cada um, uma tarefa. A tarefa em `Js2` tem uma dependência de sequência externa na instância anterior mais próxima da tarefa em `Js1`. Os dois fluxos de tarefas são definidos desta maneira:

```
SCHEDULE MY_MASTER#JS1
ON RUNCYCLE RULE1 "FREQ=DAILY;"
(AT 0800)
ON RUNCYCLE RULE2 "FREQ=WEEKLY;BYDAY=TH,FR"
(AT 0900)
:
MY_MASTER#JOB1
END

SCHEDULE MY_MASTER#JS2
ON RUNCYCLE RULE1 "FREQ=DAILY;"
(AT 1200)
FOLLOWS MY_MASTER#JS1.@ PREVIOUS
:
MY_MASTER#JOB2
AT 1500
END
```

  O fluxo de tarefas `Js1` é executado todo dia às 8h e quintas-feiras e sextas-feiras às 9h. O fluxo de tarefas `Js2` é executado todo dia às 12h e tem uma dependência externa na instância anterior mais próxima `Js1`. Quando os fluxos de tarefas são incluídos no plano, a sequência de gráficos ilustra como a dependência é resolvida:

  1. Antes das 12h nas quintas-feiras e sextas-feiras, há duas instâncias de `Js1.Job1`. O fluxo de tarefas `Js2` tem uma dependência na instância `Js1.Job1` que está planejada para execução às 9h, pois é a anterior mais próxima em termos de horário. A Figura 15 mostra o status dos dois fluxos de tarefas no plano em quintas-feiras e sextas-feiras.

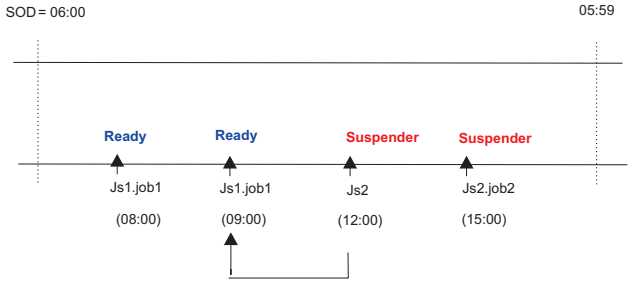


Figura 15. Critérios de Correspondência Anteriores Mais Próximos - Etapa 1: antes da 8h

  2. Em qualquer outro dia da semana, a única instância de `Js1.Job1` no plano é a planejada para execução às 8h. Nesse caso, `Js2` tem uma dependência dessa instância. Quando `Job1` for concluída com sucesso, o status de `Js2` torna-se **Pronto**. A Figura 16 na página 68 mostra o status dos dois fluxos de tarefas no plano em qualquer outro dia da semana, exceto quintas-feiras e sextas-feiras.

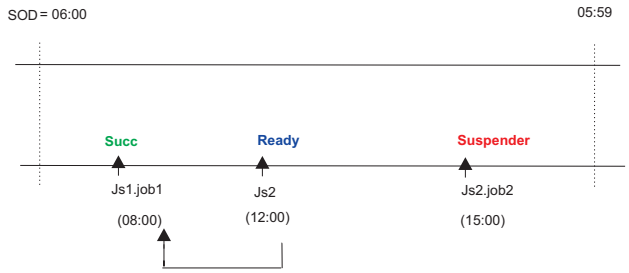


Figura 16. Critérios de Correspondência Anteriores Mais Próximos - Etapa 2: às 8h em Dias da Semana, Exceto Quintas-feiras e Sextas-feiras

3. Nas quintas-feiras e sextas-feiras às 9h, a segunda instância de `Js1.Job1` é concluída com sucesso. O fluxo de tarefas `Js2` é alterado para **Pronto**. `Js2.Job2` é retida até seu horário de início planejado. A Figura 17 mostra o status dos dois fluxos de tarefas no plano.

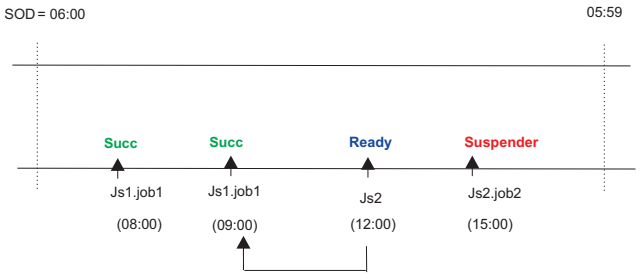


Figura 17. Critérios de Correspondência Anteriores Mais Próximos - Etapa 3: às 9h nas Quintas-feiras e Sextas-feiras

4. Às 15h, a dependência de horário de `Js2.Job2` é satisfeita e `Job2` é iniciada. A Figura 18 mostra o status dos dois fluxos de tarefas no plano às 15h.

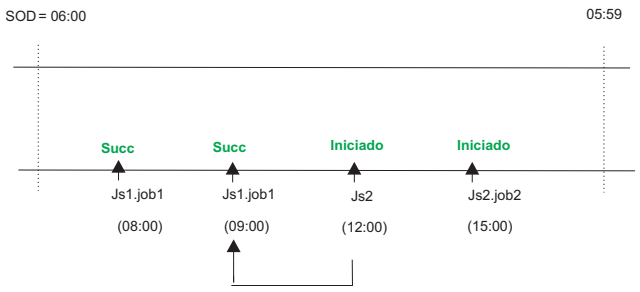


Figura 18. Critérios de Correspondência Anteriores Mais Próximos - Etapa 4: às 15h Todo Dias

Na definição de fluxo de tarefas, o ciclo de execução *Rule1* pode ser substituído pelas palavras-chave `ON EVERYDAY`.

- Neste segundo exemplo, é descrita a diferença entre o uso dos critérios de correspondência `sameday` e `closest preceding` em um plano. O fluxo de tarefas `Js1` é executado todas as Sextas-feiras às 0900, enquanto que os fluxos de tarefas `Js2` e `Js3` executam todos os Sábados às 0900. Os três fluxos de tarefas são definidos desta forma:

```
SCHEDULE
ACCOUNTING#JS1
ON RUNCYCLE RULE1 "FREQ=WEEKLY;BYDAY=FR"
:
ACCOUNTING#JOB1
```

```
 AT 0900
END

SCHEDULE ACCOUNTING#JS2
ON RUNCYCLE RULE2 "FREQ=WEEKLY;BYDAY=SA"
FOLLOWS ACCOUNTING#JS1.@ PREVIOUS
:
ACCOUNTING#JOB1
 AT 0900
END

SCHEDULE ACCOUNTING#JS3
ON RUNCYCLE RULE2 "FREQ=WEEKLY;BYDAY=SA"
FOLLOWS ACCOUNTING#JS1.@
:
ACCOUNTING#JOB1
 AT 0900
END
```

O fluxo de tarefas Js2 possui uma dependência externa na instância de precedência mais próxima de Js1, o que é resolvido como descrito no exemplo anterior. O fluxo de tarefas Js3 está definido com critérios de correspondência sameday, assim ele não possui nenhuma dependência no fluxo de tarefas Js1, pois Js1 não está configurado para ser executado no mesmo dia que Js2.

Dentro de um intervalo relativo

Neste exemplo, a instância da tarefa ou do fluxo de tarefas considerada para resolver a dependência é a mais próxima em um intervalo de tempo de sua opção, que é definido com relação ao horário em que a instância que inclui a dependência está planejada para execução. O fluxo de tarefas Js1 está planejado para execução todo dia às 15h e nas quintas-feiras também às 8h. Js2 está planejado para execução todo dia às 13h em nas quintas-feiras às 6h, como nenhum horário específico está definido no ciclo de execução, ele é planejado para o horário de início do dia. Js2 usa os critérios de intervalo relativos (-04:00 to +04:00) para determinar qual instância é usada para resolver a dependência. O intervalo é baseado no horário em que o fluxo de tarefas entra no plano. Os fluxos de tarefas são definidos da seguinte forma:

```
SCHEDULE MY_MASTER#JS1
ON RUNCYCLE RULE1 "FREQ=WEEKLY;BYDAY=TH"
(AT 0800)
ON RUNCYCLE RULE2 "FREQ=DAILY"
(AT 1500)
:
MY_MASTER#JOB1
END

SCHEDULE MY_MASTER#JS2
ON RUNCYCLE RULE3 "FREQ=WEEKLY;BYDAY=TH"
ON RUNCYCLE RULE2 "FREQ=DAILY;"
(AT 1300)
FOLLOWS MY_MASTER#JS1.@
RELATIVE FROM -0400 TO 0400
:
MY_MASTER#JOB2
AT 1300
END
```

No horário de criação do plano, **conman showjobs** produz a seguinte saída:

```
%sj @#@
                                              (Est) (Est)
CPU  Schedule SchedTime Job State Pr Start Elapse RetCode Deps
MY_MASTER#JS1  0800 11/13 ******** READY 10  (00:06)
                            JOB1  HOLD 10  (00:06)
MY_MASTER#JS1  1500 11/13 ******** READY 10  (00:06)
                           JOB1  HOLD 10  (00:06)
MY_MASTER#JS2  0600 11/13 ******** HOLD 10    JS1(0800 11/13/09).@
                            JOB2  HOLD 10(13:00)
MY_MASTER#JS2  1300 11/13 ******** HOLD 10(13:00)  JS1(1500 11/13/09).@
                            JOB2  HOLD 10(13:00)
```

A Figura 19 na página 70 mostra o status dos fluxos de tarefas no plano no início do dia na quinta-feira.

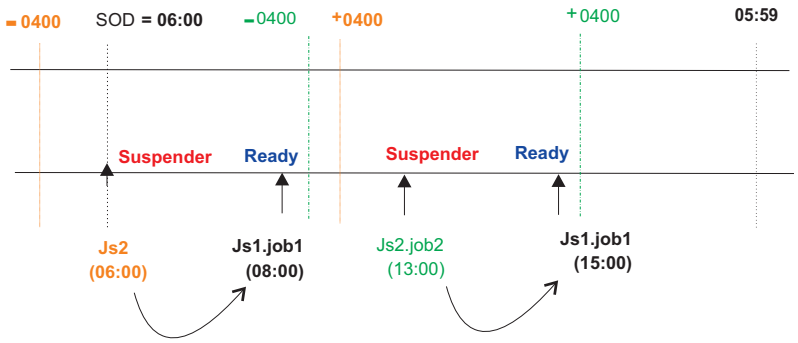


Figura 19. Critérios de Correspondência de Intervalo Relativo - no Início do Dia na Quinta-feira

A instância de Js2 planejada às 6h tem uma dependência de Js1.job1 que está planejada às 8h, dentro do intervalo relativo com base no horário planejado (6h). Js2.job2 depende da instância de Js1.job1 dentro do intervalo relativo com base no horário planejado (13h). Quando a instância de Js1.job1 inicia às 8h, o status de Js2 é alterado para Pronto. A partir desse ponto, a sequência na qual os fluxos de tarefas e tarefas são executados segue o processo típico.

Dentro de um intervalo absoluto

Neste exemplo, a instância da tarefa ou do fluxo de tarefas considerada para resolver a dependência é a mais próxima em um intervalo de tempo fixo de sua opção. O intervalo de tempo especifica o horário do dia no qual o intervalo começa e o horário do dia no qual termina, no mesmo dia que a instância que inclui a dependência ou em um dia definido com relação a essa data. Js1 está planejado para execução todo dia às 8h e nas quintas-feiras também às 7h. A tarefa Js1.job1 está planejada para execução às 9h. O fluxo de tarefas Js2 está planejado todo dias às 10h e nas quintas-feiras também no início do dia (6h) e tem uma dependência de Js1 com base no intervalo absoluto que ocorre no mesmo dia entre 6h e 11h. Os fluxos de tarefas são definidos da seguinte forma:

```
SCHEDULE MY_MASTER#JS1
ON RUNCYCLE RULE1 "FREQ=WEEKLY;BYDAY=TH"
(AT 0700)
ON RUNCYCLE RULE2 "FREQ=DAILY"
(AT 0800)
:
MY_MASTER#JOB1
AT 0900
END

SCHEDULE MY_MASTER#JS2
ON RUNCYCLE RULE3 "FREQ=WEEKLY;BYDAY=TH"
ON RUNCYCLE RULE2 "FREQ=DAILY;"
(AT 1000)
FOLLOWS MY_MASTER#JS1.@ FROM 0600 TO 1100
:
MY_MASTER#JOB2
AT 1300
END
```

No horário de criação do plano, **conman showjobs** produz a seguinte saída:

```
%sj @#@
                                                (Est) (Est)
CPU  Schedule SchedTime Job State Pr Start Elapse RetCode Deps
MY_MASTER#JS1  0700 11/13******** READY 10  (00:06)
                              JOB1  HOLD 10(09:00)(00:06)
MY_MASTER#JS1  0800 11/13 ******** READY 10  (00:06)
                              JOB1  HOLD 10(09:00)(00:06)
MY_MASTER#JS2  0600 11/13 ******** HOLD 10    JS1(0700 11/13/09).@
                              JOB2  HOLD 10(15:00)
MY_MASTER#JS2  1000 11/13 ******** HOLD 10(10:00)  JS1(0800 11/1309).@
                              JOB2  HOLD 10(15:00)
```

A Figura 20 mostra o status dos fluxos de tarefas no plano no início do dia na quinta-feira.
Às 9h, `Js1.job1` é iniciado, e às 10h, a dependência é liberada e `Js2` fica

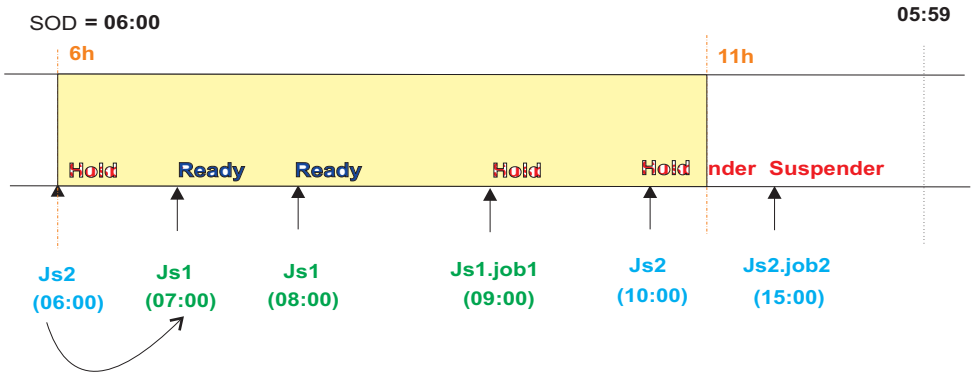


Figura 20. Critérios de Correspondência de Intervalo Absoluto - no Início do Dia na Quinta-feira

pronto. A partir desse ponto, a sequência é a mesma que a descrita nos critérios de correspondência anteriores.

Plano de Produção

Após ter criado ou atualizado o plano de pré-produção, o Tivoli Workload Scheduler conclui as informações armazenadas no plano de pré-produção com as informações armazenadas no banco de dados sobre as operações a serem desempenhadas no período de tempo selecionado e os outros objetos de planejamento envolvidos ao processar o plano e copia-o em um novo arquivo `Symphony`. Também inclui nesse arquivo as dependências entre planos, como fluxos de tarefas com saldo transportado do plano de produção já processado e arquiva o arquivo `Symphony` antigo no diretório `schedlog`.

No final deste processo, o novo arquivo `Symphony` conterá todas as informações que implementam o novo plano de produção e, além disso, todos os dados do plano são replicados no banco de dados para facilitar a consulta a partir do Dynamic Workload Console e das APIs.

Uma cópia do novo arquivo `Symphony` é distribuída a todas as estações de trabalho envolvidas na execução de tarefas ou fluxos de tarefas para esse plano de produção.

No arquivo de segurança, a autorização do usuário necessária para gerar o plano de produção é a palavra-chave de acesso *build* nos arquivos `prodsked` e `Symphony`.

Nota: Para evitar a execução com falta de espaço em disco, lembre-se que cada instância da tarefa ou do fluxo de tarefas aumenta o tamanho do arquivo `Symphony` em 512 bytes.

Para obter informações sobre como gerar o plano de produção, consulte “Criando e estendendo o plano de produção” na página 79.

Entendendo as Opções de Transporte de Saldo

Os fluxos de tarefas têm o saldo transportado quando o plano de produção é gerado. Como o fluxo de tarefas tem o saldo transportado depende de:

- A palavra-chave **carryforward** no fluxo de tarefas. Consulte "carryforward" na página 221.
- A opção global **enCarryForward**. Consulte o *Guia de Administração do IBM Tivoli Workload Scheduler*.
- A palavra-chave da linha de comandos **stageman -carryforward**. Consulte "O comando stageman" na página 95.
- A opção global **carryStates**. Consulte o *Guia de Administração do IBM Tivoli Workload Scheduler*.

A Tabela 12 mostra como as opções globais de transporte de saldo funcionam juntas.

Tabela 12. Configurações das Opções Globais do Saldo Transportado

Opções Globais	Operação de transporte
enCarryForward=all carryStates=()	Os fluxos de tarefas têm o saldo transportado somente se não tiverem sido concluídos. Todos os jobs são transportados com os fluxos de job. Essa é a configuração padrão.
enCarryForward=no	Nenhum fluxo de job é transportado. Se esta opção estiver configurada como no, a execução de tarefas é transferida para o fluxo de tarefas USERJOBS.
enCarryForward=yes carryStates=(*states*)	Os fluxos de tarefa tem o saldo transportado somente se tiverem ambas tarefas nos estados especificados e a palavra-chave **carryforward** configurada na definição do fluxo de tarefa. Apenas os jobs nos estados especificados são transportados com os fluxos de jobs.
enCarryForward=yes carryStates=()	Os fluxos de tarefas têm o saldo transportado somente se eles não tiverem sido concluídos e tiverem a palavra-chave **carryforward** configurada na definição do fluxo de tarefas. Todos os jobs são transportados com os fluxos de job.
enCarryForward=all carryStates=(*states*)	Os fluxos de job são transportados apenas se tiverem jobs nos estados especificados. Apenas os jobs nos estados especificados são transportados com os fluxos de job.

A Tabela 13 mostra o resultado da configuração do saldo transportado com base em como a opção global **enCarryForward** e as palavras-chave **stageman -carryforward** são configuradas.

Tabela 13. Configurações Resultantes do Transporte de Saldo

enCarryForward	stageman -carryforward	Configuração resultante do transporte de saldo
NÃO	SIM	NÃO
NÃO	TODOS	NÃO
SIM	NÃO	NÃO
TODOS	NÃO	NÃO
TODOS	SIM	TODOS
SIM	TODOS	TODOS
SIM	SIM	SIM

A opção transportar saldo configurada na definição do fluxo de trabalho é persistente. Isso significa que um fluxo de tarefas malsucedido que estiver marcado como **carryforward**, continua a ter o saldo transportado até que ocorra uma das seguintes situações:

- Ele seja finalizado em um estado SUCC
- Sua hora UNTIL seja alcançada
- Ele seja cancelado.

A convenção de nomenclatura do fluxo de tarefas redirecionado é afetada pelo valor designado à opção global ***enLegacyId***. Para obter informações adicionais sobre as diferentes configurações para esta opção, consulte "Customizando o Gerenciamento de Planos Utilizando Opções Globais" na página 75.

Nota: Independentemente de como as opções de transporte de saldo são configuradas, os fluxos de tarefas que não contêm tarefas não têm o saldo transportado.

Se você configurar **carryStates=(succ)** e **enCarryForward=all** ou **enCarryForward=yes** , a próxima vez que você executar **JnextPlan** haverá uma falta de alinhamento entre o plano de pré-produção e o novo arquivo Symphony. Isso ocorre porque o plano de pré-produção não contém as instâncias dos fluxos de tarefas que foram encerrados com êxito, mas o novo arquivo Symphony contém. O resultado dessa falta de alinhamento é que as dependências não são resolvidas de acordo com as instâncias do fluxo de tarefas que tiveram o saldo transportado com êxito, pois não existem mais no plano de pré-produção.

A decisão de transportar uma tarefa repetitiva, que é uma tarefa que contém uma configuração de tempo **every** em sua definição, ou uma cadeia de tarefas reexecutadas baseia-se no estado de sua execução mais recente. Apenas a primeira e a última tarefas da cadeia são transportadas.

Plano de Experiência

Um plano trial é uma projeção do que seria um plano de produção se cobrisse um período de tempo mais longo. Por exemplo, se você gerar um plano de produção com cobertura de dois dias, mas deseja saber qual seria o plano se cobrisse três dias, será possível gerar um plano de experiência.

Estas são as características de um plano trial:

- Sua data de início corresponde:
  - À data de início do plano de pré-produção.
  - À data de encerramento do plano de produção.
- É baseado nas informações estáticas armazenadas no plano de pré-produção atual.
- Não pode ser executado para gerenciar produção.
- Pode ser gerenciado por usuários com acesso *build* no tipo de objeto do arquivo **trialsked** configurado no arquivo de segurança no gerenciador de domínio principal.
- Produz um arquivo armazenado no diretório schedlog com estas propriedades:
  - O mesmo formato que o arquivo Symphony.
  - O nome do arquivo é iniciado com um T.

As gerações de plano de experiência podem resultar na extensão do horário de encerramento do plano de pré-produção. Isto depende das configurações das opções globais minLen e maxLen. Quando isto acontece, o banco de dados é bloqueado e desbloqueado somente quando a operação é concluída.

Não há nenhuma restrição sobre o período de tempo selecionado para um plano trial, mas o tamanho do arquivo resultante que contém todas as informações do plano trial deve ser levado em consideração.

Como o plano trial é baseado nas informações estáticas armazenadas no plano de pré-produção, não leva em consideração quaisquer atualizações dinâmicas feitas no arquivo Symphony enquanto o plano de produção está sendo processado, de forma que todos os fluxos de tarefas contidos estejam em um destes dois estados:

HOLD
: Se dependerem de outros fluxos de tarefas ou se o horário de início for posterior ao início do horário do plano.

READY
: Se estiverem livres das dependências e sua hora de início tiver passado.

As operações que podem ser executadas em um plano trial no gerenciador de domínio principal são:

creation
Utilizada para criar um plano trial para ter uma visão geral da produção quando um plano de produção ainda não existir.

extensão
Utilizada para criar um plano trial da extensão do plano de produção atual para ter uma visão geral de como a produção evolui no futuro.

Para obter informações sobre como criar ou estender um plano trial, consulte "Linha de Comandos Planman" na página 82.

Plano de Previsão

O *plano de previsão* é uma projeção do que seria o plano de produção em um espaço de tempo escolhido. Por exemplo, se você tiver gerado um plano de produção que cubra dois dias e quiser saber como o plano seria na próxima semana, é possível gerar um plano de previsão.

Estas são as características de um plano de previsão:

- Cobre qualquer espaço de tempo, futuro, passado ou até parcialmente sobrepondo o período de tempo coberto pelo plano de produção atual.
- É baseado em um plano de pré-produção de amostra que cobre o mesmo período de tempo selecionado para o plano de previsão. Esse plano de pré-produção de amostra é excluído após o plano de previsão ser criado.
- Não pode ser executado para gerenciar produção.
- Pode ser gerenciado por usuários com acesso *build* no tipo de objeto do arquivo **trialsked** configurado no arquivo de segurança no gerenciador de domínio principal.
- Produz um arquivo armazenado no diretório `schedlog` com estas propriedades:
  - O mesmo formato que o arquivo `Symphony`.
  - O nome do arquivo é iniciado com um F.
- Quando o garantia de serviço de carga de trabalho está ativado, ele pode calcular o horário de início previsto de cada tarefa do fluxo de tarefas. É possível ativar e desativar esse recurso usando a opção global **enForecastStartTime**. O Tivoli Workload Scheduler calcula a duração de execução média para cada tarefa com base em todas as execuções anteriores. Para planos complexos, ativar esse recurso poderia afetar de forma negativa o tempo que leva para gerar o plano de previsão.

Ao criar um plano de previsão, o banco de dados é bloqueado e somente desbloqueado quando a operação é concluída.

Não há nenhuma restrição sobre o período de tempo selecionado para gerar um plano de previsão, mas o tamanho do arquivo resultante que contém todas as informações do plano de previsão deve ser levado em consideração.

Como o plano de previsão é baseado nas informações estáticas armazenadas no banco de dados, não leva em consideração quaisquer atualizações dinâmicas feitas no arquivo `Symphony` enquanto o plano de produção ou o plano de pré-produção está sendo processado, de forma que todos os fluxos de tarefas contidos estejam em um destes dois estados:

HOLD
Se dependerem de outros fluxos de tarefas ou se o horário de início for posterior ao início do horário do plano.

READY
Se estiverem livres das dependências e sua hora de início tiver passado.

A operação que pode ser executada em um plano de previsão no gerenciador de domínio principal é:

creation
É utilizada para criar um plano de previsão para ter uma visão geral da produção em um espaço de tempo escolhido.

Para obter informações sobre como criar um plano de previsão, consulte "Linha de Comandos Planman" na página 82.

Customizando o Gerenciamento de Planos Utilizando Opções Globais

É possível customizar alguns critérios para o Tivoli Workload Scheduler utilizar ao gerenciar planos configurando opções específicas no gerenciador de domínio principal utilizando o programa de linha de comandos **optman**. É necessário gerar o plano novamente para ativar as novas configurações. As opções que podem ser customizadas são:

Propriedades que afetam a geração do plano de pré-produção:

minLen
É a duração mínima, calculada em dias, do plano de pré-produção, que é deixada como um buffer após o final do novo plano de produção gerado. O valor designado a essa opção é utilizado quando o script **UpdateStats** é executado a partir do **JnextPlan**. O valor pode ser de 7 a 365 dias. O padrão é 8 dias.

maxLen
É a duração máxima, calculada em dias, do plano de pré-produção, que é deixada como um buffer após o final do novo plano de produção gerado. O valor pode ser de 8 a 365 dias. O padrão é 8 dias.

Se os valores de `minLen` e `maxLen` forem iguais, o plano de pré-produção é atualizado durante a fase `MakePlan`. Em geral, o valor de `maxLen` deve exceder o valor de `minLen` em pelo menos 1 dia, de forma que o plano de pré-produção possa ser atualizado durante a fase `UpdateStats`.

Propriedades que afetam a geração ou a extensão do plano de produção:

startOfDay
Representa o horário de início do dia de processamento do Tivoli Workload Scheduler no formato de 24 horas: hhmm (0000-2359). a configuração padrão é 0000.

enCarryForward
Essa é uma opção que afeta como o comando **stageman** transporta o saldo de fluxos de tarefas. Sua configuração determina se os fluxos de tarefa que não foram concluídos têm ou não o saldo transportado do antigo plano de produção para o novo. As configurações disponíveis para *enCarryForward* são **yes, no** e **all**. A configuração padrão é **all.**

carryStates

Essa é uma opção que afeta como o comando **stageman** gerencia tarefas nos fluxos de tarefas com saldo transportado. Sua configuração determina, com base em seu estado, as tarefas a serem incluídas nos fluxos de tarefa que têm o saldo transportado. Por exemplo, se:

```
carryStates='abend exec hold'
```

então, todas as tarefas que estão no estado `abend`, `exec` ou `hold` são incluídas nos fluxos de tarefa com saldo transportado. A configuração padrão é:

```
carryStates=null
```

isso significa que todas as tarefas são incluídas, independentemente de seus estados.

enCFInterNetworkDeps

Essa é uma opção que afeta como o comando **stageman** gerencia dependências da interligação de redes. Digite **yes** para que todos os fluxos de tarefa `EXTERNAL` tenham o saldo transportado. Digite **no** para desativar completamente a função transportar saldo para as dependências da interligação de redes. A configuração padrão é **yes**.

enCFResourceQuantity

Essa é uma opção que afeta como o comando **stageman** gerencia recursos. Quando o plano de produção é estendido, ocorre uma das seguintes situações:

- Um recurso não utilizado por nenhum dos fluxos de tarefa com saldo transportado do plano de produção anterior é referido por novas instâncias de tarefa ou de fluxo de tarefa incluídas no novo plano de produção. Nesse caso, a quantidade do recurso é obtida da definição do recurso armazenada no banco de dados.
- Um recurso utilizado para um ou mais fluxos de tarefa com saldo transportado do plano de produção anterior não é referido por instâncias de tarefa ou de fluxo de tarefa incluídas no novo plano de produção. Nesse caso, a quantidade do recurso é obtida do antigo arquivo `Symphony`.
- Um recurso utilizado por um ou mais fluxos de tarefa com saldo transportado do plano de produção anterior é referido por instâncias de tarefa ou de fluxo de tarefa incluídas no novo plano de produção. Nesse caso, a quantidade do recurso que é levada em consideração é baseada no valor designado à opção *enCFResourceQuantity*:

  Se *enCFResourceQuantity* **estiver configurada para YES**
  A quantidade do recurso é obtida do antigo arquivo `Symphony`.

  Se *enCFResourceQuantity* **estiver configurada para NO**
  A quantidade do recurso é obtida da definição do recurso armazenada no banco de dados.

A configuração padrão é **yes**.

enEmptySchedsAreSucc

Essa opção rege o comportamento dos fluxos de tarefas que não contêm tarefas. As configurações disponíveis são:

sim Os fluxos de tarefas que não contêm tarefas são marcados como SUCC à medida que suas dependências são resolvidas.

não Os fluxos de tarefas que não contêm tarefas permanecem no estado READY.

enPreventStart

Essa é uma opção para gerenciar, para um plano de produção de vários dias, quaisquer fluxos de tarefas sem uma restrição de tempo **at** configurada. É utilizada para evitar que as instâncias do fluxo de tarefas que não são dependentes de tempo sejam iniciadas todas ao mesmo tempo à medida que o plano de produção é criado ou estendido. As configurações disponíveis são:

sim Um fluxo de tarefas não pode ser iniciado antes de *startOfDay* do dia especificado em seu horário planejado mesmo que livre de dependências.

não Um fluxo de tarefas pode ser iniciado imediatamente quando o plano de produção é iniciado se todas as suas dependências estiverem resolvidas.

enLegacyId

Essa é uma opção que afeta como os fluxos de tarefas são denominados no plano. Sua função é manter a consistência ao identificar fluxos de tarefas no plano em ambientes combinados com versões do Tivoli Workload Scheduler anteriores à 8.3 gerenciadas pelos gerenciador de domínio principals versão 8.4. Esta opção não é suportada pelo catálogo de Autoatendimento, que o ignora mesmo se o valor estiver configurado como YES. O valor designado a esta opção é lido quando o plano de produção é criado ou estendido ou ao enviar fluxos de tarefas na produção utilizando **conman**. As configurações disponíveis são:

sim O *jobstream_id* do fluxo de tarefas denominado *jobstream_name* é configurado para *jobstream_nameN*, e em que *N* é um número incremental designado por um contador interno; ele será configurado para *null* se apenas uma instância desse fluxo de tarefas existir no plano.

Se o fluxo de tarefas denominado *jobstream_name* for transportado, seu identificador será configurado para *CFjobstream_nameN*.

Essa configuração será útil para manter a consistência ao gerenciar fluxos de tarefas, mesmo aquelas transportadas, utilizando **conman** quando efetuado logon em um agente do Tivoli Workload Scheduler 8.2.x em uma rede do Tivoli Workload Scheduler com um gerenciador de domínio principal 8.4.

Especificamente, se o período de produção for um dia e não forem submetidas várias instâncias do mesmo fluxo de tarefas, a compatibilidade com versões anteriores, ao gerenciar fluxos de tarefas na produção de um agente Tivoli Workload Scheduler versão 8.2.x, está concluída.

não O identificador do fluxo de tarefas *jobstream_id* é gerado conforme descrito em “showjobs” na página 411. Os fluxos de tarefas transportados mantêm seus nomes e

identificadores originais e relatam entre chaves {} a data de quando tiveram o saldo transportado.

logmanSmoothPolicy
Essa é uma opção que afeta como o comando **logman** manipula as estatísticas e o histórico. Configura o fator de peso que favorece a tarefa mais recente ao calcular o tempo de execução normal (médio) para uma tarefa. Isso é expresso como porcentagem. A configuração padrão é **-1**, o que significa que essa propriedade não está ativada.

logmanMinMaxPolicy
Essa opção define como os tempos de execução mínimo e máximo da tarefa são registrados e relatados por **logman**. As configurações disponíveis para a opção *logmanMinMaxPolicy* são:

elapsedtime
Os tempos de execução e as datas máximo e mínimo que são registrados são baseados somente no tempo decorrido de uma tarefa. O tempo passado, expresso em minutos, é bastante afetado pela atividade do sistema. Inclui o período de tempo de CPU utilizado por uma tarefa e o tempo que a tarefa precisou aguardar por outros processos para liberar a CPU. Em períodos de alta atividade do sistema, por exemplo, um job pode ter um longo tempo decorrido e não utilizar mais tempo de CPU do que em períodos de baixa atividade do sistema. Os valores são atualizados apenas se a execução de job mais recente tiver um tempo decorrido maior que o máximo existente, ou menor que o mínimo existente.

cputime
Os tempos de execução e as datas máximo e mínimo que são registrados são baseados somente no tempo de CPU de uma tarefa. O tempo de CPU é uma medida, expressa em segundos, do tempo real que uma tarefa utilizou a CPU e não inclui os intervalos em que a tarefa estava aguardando. Os valores são atualizados apenas se a execução de job mais recente tiver um tempo de CPU maior que o máximo existente, ou menor que o mínimo existente.

both Os valores de tempo decorrido e de tempo de CPU são atualizados de forma independente para indicar seus extremos máximo e mínimo, mas as datas de execução correspondem somente aos valores de tempo decorridos. Nenhum registro é mantido, neste caso, das datas de execução para os tempos de CPU máximo e mínimo.

A configuração padrão é **both**.

enTimeZone
Ao configurar a opção, você ativa ou desativa o gerenciamento de fusos horários na rede do Tivoli Workload Scheduler. As configurações disponíveis para a opção *enTimeZone* são:

não Desativar o gerenciamento de fuso horário. Isso significa que os valores designados a todas as palavras-chave **timezone** nas definições são ignorados.

sim Ativar o gerenciamento de fuso horário. Isso significa que

os valores designados às configurações de **timezone** são utilizados para calcular a hora em que os fluxos de tarefas e as tarefas serão executados nas estações de trabalho de destino.

Consulte "Ativando o Gerenciamento de Fuso Horário" na página 597 para obter informações adicionais sobre a variável *enTimeZone*.

enLegacyStartOfDayEvaluation
Essa opção afeta a maneira que a variável *startOfDay* é gerenciada na rede do Tivoli Workload Scheduler. Essa opção requer que a variável *enTimeZone* configurada para **yes** se torne operacional. As configurações disponíveis para a opção *enLegacyStartOfDayEvaluation* são:

não O valor designado à opção *startOfDay* no gerenciador de domínio principal não é convertido no fuso horário local configurado em cada estação de trabalho na rede.

sim O valor designado à opção *startOfDay* no gerenciador de domínio principal é convertido no fuso horário local configurado em cada estação de trabalho na rede.

Consulte "Como o Tivoli Workload Scheduler Gerencia Fusos Horários" na página 598 para obter informações adicionais sobre a variável *enLegacyStartOfDayEvaluation*.

Para obter informações sobre como configurar opções utilizando o programa da linha de comandos **optman**, consulte o *Guia de Administração do IBM Tivoli Workload Scheduler*.

Criando e estendendo o plano de produção

Todo o processo de movimentação de um plano de produção antigo para um novo, incluindo sua ativação na rede do Tivoli Workload Scheduler, é gerenciada pelo script **JnextPlan**. É possível executar **JnextPlan** a qualquer momento durante o dia de processamento. O novo plano de produção que é gerado é ativado imediatamente nas estações de trabalho de destino, independentemente da hora configurada na variável ***startOfDay***. Quando o comando **JnextPlan** é executado, a variável *$MANAGER* é gerenciada da seguinte forma:

- A variável é resolvida se a estação de trabalho for um agente tolerante a falhas de uma versão anterior à 8.6.
- A variável é deixada sem ser resolvida para estações de trabalho do agente tolerante a falhas versão 8.6.

Ao executar o script **JnextPlan**, os processos da estação de trabalho são parados e reiniciados em todas as estações de trabalho na rede do Tivoli Workload Scheduler. Para obter mais informações sobre processos de estação de trabalho, consulte Capítulo 2, "Entendendo Processos e Comandos Básicos", na página 31.

O script **JnextPlan** pode ser executado apenas a partir do gerenciador de domínio principal. Ele usar os parâmetros de conexão padrão definidos nos arquivos `localopts` ou `useropts` (consulte "Configurando Opções para Utilização das Interfaces com o Usuário" na página 53). Se quiser executar **JnextPlan** utilizando diferentes configurações de parâmetros de conexão, você pode editar o script **MakePlan** e modificar a chamada para a instrução **planman** conforme descrito em "Linha de Comandos Planman" na página 82.

O script **JnextPlan** é composto pela seguinte seqüência de comandos e scripts especializados, cada uma gerenciando um aspecto específico da geração do plano de produção:

conman startappserver
Este comando é chamado para iniciar oWebSphere Application Server se ele ainda não estiver em execução.

MakePlan
Esse script herda os sinalizadores e os valores designados a eles de **JnextPlan**. Sua sintaxe é:

MakePlan [**-from** *mm/dd/[yy]yy[hh[:]mm[**tz** | **timezone** tzname]]*] {**-to** *mm/dd/[yy]yy[hh[:]mm[**tz** | **timezone** tzname]]* | **-for** *[h]hh[:]mm* [**-days** *n*] | **-days** *n*}

MakePlan chama internamente a linha de comandos **planman**. **MakePlan** executa as seguintes ações:

1. Cria um novo plano e estende o plano atual e armazena as informações em um plano de produção intermediário que contém:
   - Todos os objetos de planejamento (tarefas, fluxos de tarefas, calendários, avisos, recursos, estações de trabalho, domínios, arquivos, usuários, dependências) definidos no período de tempo selecionado.
   - Todas as dependências entre novas instâncias de tarefas e de fluxos de tarefa e as tarefas e os fluxos de tarefa existentes no plano de produção anterior.
   - Todas as solicitações de ligação cujo tempo planejado está incluído no período de tempo selecionado.
2. Exclui todas as solicitações de ligação no estado final.
3. Imprime relatórios de pré-produção.

SwitchPlan
Esse script chama o comando **stageman** internamente. Para obter mais informações, consulte "O comando stageman" na página 95. **SwitchPlan** executa as seguintes ações:

1. Para processos do Tivoli Workload Scheduler.
2. Gera o novo arquivo `Symphony` iniciando no plano de produção intermediário criado pelo **MakePlan**.
3. Arquiva o arquivo de plano antigo com a data e hora atuais no diretório `schedlog`.
4. Cria uma cópia do arquivo `Symphony` para distribuir para as estações de trabalho.
5. Reinicia processos do Tivoli Workload Scheduler que distribuem a cópia do arquivo `Symphony` para os destinos da estação de trabalho para executar as tarefas no plano.

Nota: Certifique-se de que nenhum comando **conman start** seja executado enquanto o plano de produção está sendo processado.

CreatePostReports
Esse script imprime os relatórios de pós-produção.

UpdateStats
Esse script chama o comando **logman** internamente. Para obter mais informações, consulte "O comando stageman" na página 95. **UpdateStats** executa as seguintes ações:

1. Registra estatísticas de tarefas.
2. Verifica as políticas e, se necessário, estende o plano de pré-produção.
3. Atualiza o plano de pré-produção relatando os estados de instância do fluxo de tarefas.

Para obter mais informações sobre como utilizar o script **JnextPlan**, consulte "JnextPlan".

Nota: Para obter informações sobre os cenários específicos que podem requerer a customização de **JnextPlan** , consulte o *Guia de Administração do IBM Tivoli Workload Scheduler*

JnextPlan

O script **JnextPlan** é usado para gerenciar o processo inteiro de mudança de um plano de produção antigo para um novo (Symphony), incluindo sua ativação na rede do Tivoli Workload Scheduler. Toda vez que **JnextPlan** é executada, todas as estações de trabalho são paradas e interrompidas.

Ao executar o comando **JnextPlan** um arquivo de log da tarefa é criado no diretório <*TWS_INST_DIR*>\TWS\stdlist\<*DATE*>, em que <*TWS_INST_DIR*> é o diretório de instalação do Tivoli Workload Scheduler e <*DATE*> é a data em que o script foi executado.

Autorização

É possível executar o comando **JnextPlan** a partir de um shell de prompt de comandos no gerenciador de domínio principal se você for um dos seguintes usuários:

- O usuário *TWS_user* para o qual você instalou o produto nessa máquina, se não desativado pelas configurações definidas no arquivo de segurança.
- Raiz em sistemas operacionais UNIX ou Administrador nos sistemas operacionais Windows, se não estiver desativado pelas configurações que são definidas no arquivo de segurança.

Sintaxe

JnextPlan
[**-V** | **-U**] |
[**-from** *mm/dd/[yy]yy[hh[:]mm[**tz** | **timezone** tzname]]*]
{**-to** *mm/dd/[yy]yy[hh[:]mm[**tz** | **timezone** tzname]]* |
-for *[h]hh[:]mm* [**-days** *n*] | **-days** *n*}
[**-noremove**]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

-from Configura a hora de início do plano de produção. O formato da data é especificado no arquivo `localopts`; onde *hhmm* identifica as horas e os minutos e *tz* é o fuso horário. Esse sinalizador é utilizado somente se um plano de produção não existir. Se o argumento **-from** não for especificado, o valor padrão será "`today +startOfDay`".

Se nenhum fuso horário for especificado, o fuso horário GMT será utilizado por padrão.

-to É o horário de encerramento do plano de produção. O formato para a data é o mesmo que o utilizado para o argumento **-from**. O argumento **-to** é mutuamente exclusivo com os argumentos **-for** and **-days**.

Se nenhum fuso horário for especificado, o fuso horário GMT será utilizado por padrão.

-for É a extensão do plano expressa em tempo. O formato é *hhhmm*, em que *hhh* são as horas e *mm* são os minutos. O argumento **-for** é mutuamente exclusivo com **-to**.

-days *n*
É o número de dias pelo qual você deseja criar ou estender o plano de produção. O parâmetro**-days** é mutuamente exclusivo com o parâmetro**-to**.

-noremove
Assegura que as instâncias do fluxo de tarefas concluídas não são removidas do novo plano de produção.

Se nenhum argumento **-to**, **-for** ou **-days** for especificado, a duração do plano de produção padrão será de um dia.

JnextPlan -for 0000

O comando `JnextPlan -for 0000` estende em 0 horas e 0 minutos o plano de produção e inclui no plano de produção (Symphony) a estação de trabalho recém-criada, o usuário e as definições de calendário no banco de dados. Ele também remove todas as instâncias do fluxo de tarefa concluídas com êxito.

Se você utilizar o comando `JnextPlan -for 0000 -noremove`, nenhuma instância do fluxo de tarefas, concluída com êxito no Symphony, será removida.

A definição de opção global `enCarryForward` especifica se os fluxos de tarefas não concluídos são transportados do plano de produção antigo para o novo. Certifique-se de que a opção `enCarryForward` esteja configurada para `ALL` antes de executar o comando para ter todas as instância do fluxo de tarefa incompletas no novo plano de produção ou usar a opção `-noremove`.

Exemplo

Supondo que o valor designado para *startOfDay* seja 0h e que o formato de data configurado no arquivo `localopts` seja *mm/dd/yyyy*, se os valores configurados forem **-from** `07/05/2011` e **-to** `07/07/2011`, então, o plano será criado para se estender pelo período de tempo de 05/07/2011 às 0h a 06/07/2011 às 23h59 e não até 07/07/2011 às 23h59.

Linha de Comandos Planman

A linha de comandos **planman** é utilizada para gerenciar planos de produção *intermediários*, planos de *experiência* e planos de *previsão*. Ela também é usada para obter informações sobre o plano de produção atualmente ativo, para desbloquear as entradas do banco de dados bloqueadas pelos processos de gerenciamento de plano, para implementar as regras de planejamento de evento e para replicar os dados do plano no banco de dados. O comando é executado no gerenciador de domínio principal. Utilize a seguinte sintaxe ao executar o **planman**:

planman -U

planman -V

planman [*connection_parameters*] *command*

em que:

-U Exibe informações de uso do comando e sai.

-V Exibe a versão do comando e sai.

connection_parameters

Se estiver usando **planman** a partir do gerenciador de domínio principal, os parâmetros de conexão foram configurados na instalação e não precisam ser fornecidos, a menos que você não deseje usar os valores padrão.

Se estiver usando **planman** a partir do cliente da linha de comandos em outra estação de trabalho, os parâmetros de conexão podem ser fornecidos por um ou mais destes métodos:

- Armazenados no arquivo `localopts`
- Armazenados no arquivo `useropts`
- Fornecidos ao comando em um arquivo de parâmetro
- Fornecidos ao comando como parte da sequência de caracteres de comando

Para obter uma visão geral dessas opções, consulte "Configurando Opções para Utilização das Interfaces com o Usuário" na página 53. Para obter detalhes integrais dos parâmetros de configuração, consulte o tópico sobre como configurar o acesso do cliente de linha de comandos no *Tivoli Workload Scheduler: Administration Guide*.

comando

Representa o comando executado para gerenciar planos usando a interface **planman**. Essas são as ações que podem ser executadas com os planos:

- "Criando um Plano de Produção Intermediário" na página 84
- "Criando um Plano Intermediário para uma Extensão de Plano" na página 85
- "Recuperando informações sobre o plano de produção" na página 86
- "Criando um Plano Trial" na página 87
- "Criando um Plano Trial de uma Extensão de Plano de Produção" na página 88
- "Criando um Plano de Previsão" na página 89
- "Desbloqueando o Plano de Produção" na página 91
- "Removendo o Plano de Pré-Produção" na página 92
- "Reconfigurando o plano de produção" na página 91
- "Replicando os Dados do Plano no Banco de Dados" na página 93
- "Monitorando a Replicação dos Dados do Plano no Banco de Dados" na página 94

Você também pode utilizar **planman** para implementar as regras de eventos de planejamento. O comando é explicado em: "Regras de Implementação" na página 90. Consulte as subseções relacionadas para obter detalhes adicionais sobre esses comandos.

Criando um Plano de Produção Intermediário

O **planman** com a opção **crt** é chamado a partir do comando **JnextPlan** em uma destas duas situações:

- A primeira vez que o comando **JnextPlan** é executado após ter instalado o produto.
- Ao gerar um plano de produção após ter reconfigurado o plano de produção utilizando o comando **ResetPlan**.

O resultado da execução desse comando é a criação de um novo plano de produção intermediário, denominado`Symnew`, que cobre todo o tempo que o novo plano de produção que está sendo gerado cobrirá. A seguinte sintaxe é utilizada:

planman [*connection_parameters*] **crt**

[**-from** *mm/dd/*[*yy*]*yy* [*hh*[*:*]*mm* [**tz** | **timezone** *tzname*]]]

{**-to** *mm/dd/*[*yy*]*yy*[*hh*[*:*]*mm*[***tz*** | ***timezone*** *tzname]]* |

-for *[h]hh[:]mm* [**-days** *n*] |

-days *n*}

em que:

connection_parameters
Define as configurações a serem utilizadas ao estabelecer a conexão utilizando HTTP ou HTTPS através do WebSphere Application Server com o gerenciador de domínio principal. Para obter mais informações, consulte "Linha de Comandos Planman" na página 82.

-from Configura a hora de início do novo plano de produção.

Se o argumento **-from** for omitido, então:

- A data padrão é *today*.
- A hora padrão é o valor configurado na opção global ***startOfDay*** utilizando**optman** no gerenciador de domínio principal.

-to É o novo horário de encerramento do plano de produção. O formato para a data é o mesmo que o utilizado para o argumento **-from**. O argumento **-to** é mutuamente exclusivo com os argumentos **-for** and **-days**.

-for É a extensão do plano expressa em tempo. O formato é *hhhmm*, em que *hhh* são as horas e *mm* são os minutos. O argumento **-for** é mutuamente exclusivo com o argumento **-to**.

-days *n*
É o número de dias para o qual você deseja criar o plano de produção. O argumento **-days** é mutuamente exclusivo com o argumento **-to**.

Nota:

1. Certifique-se de executar o comando **planman** a partir do comando **JnextPlan**.
2. O formato utilizado para a data depende do valor designado para a variável ***date format*** especificada no arquivo `localopts`.

Se nenhum argumento **-to**, **-for** ou **-days** for especificado, a duração do plano de produção padrão será de um dia.

Estes são alguns exemplos de utilização do comando **planman**, supondo que o formato de data configurado no arquivo `localopts` seja dd/mm/aaaa:

1. Este comando cria o plano de produção de 21/03/2011 às 23h07 até 22/03/2011 às 23h06 no fuso horário local:

   ```
   planman crt –from 21/03/05 2307
   ```

2. Este comando cria o plano de produção de 21/03/2011 às 9h até 21/03/2011 às 15h:

   ```
   planman crt –from 03/21/2011 0900 for 0600
   ```

3. Se hoje for 21/03/05 e o valor configurado para a variável *startOfDay* armazenada no banco de dados for 0600, este comando cria o plano de produção de 21/03/2011 às 6h paté 25/03/2011 às 5h59:

   ```
   planman crt –to 03/25/2011
   ```

4. Este comando cria um plano de produção de 21/03/2011 às 18h05 até 24/03/2011 às 23h no fuso horário Europe\Paris:

   ```
   planman crt –from 03/21/2011 1805 tz Europe\Rome
               –to 03/24/2011 2300 tz Europe\Rome
   ```

Criando um Plano Intermediário para uma Extensão de Plano

O comando **planman** com a opção **ext** é chamado a partir do comando **JnextPlan** quando:

- **JnextPlan** é chamado.
- Um plano de produção, representado pelo arquivo `Symphony` no gerenciador de domínio principal, já existe.

O resultado da execução deste comando é a criação de um novo plano de produção intermediário, denominado`Symnew`, cobrindo o tempo extra pelo qual o novo plano de produção que está sendo gerado se estenderá. A seguinte sintaxe é utilizada:

planman [*connection_parameters*] **ext**

{**-to** *mm/dd/[yy]yy[hh[:]mm[**tz** | **timezone** tzname]]* |

-for *[h]hh[:]mm* [**-days** *n*] |

-days *n*}

em que:

connection_parameters
: Define as configurações a serem utilizadas ao estabelecer a conexão utilizando HTTP ou HTTPS através do WebSphere Application Server com o gerenciador de domínio principal. Para obter mais informações, consulte "Linha de Comandos Planman" na página 82.

-to
: Configura o horário de encerramento do plano de produção estendido. O argumento **-to** é mutuamente exclusivo com os argumentos **-for** and **-days**.

-for
: Configura a duração da extensão do plano de produção. O formato é *hhhmm*, em que *hhh* são as horas e *mm* são os minutos. O argumento **-for** é mutuamente exclusivo com o argumento **-to**.

-days *n*
: Configura o número de dias para o qual você deseja estender o plano de produção. O argumento **-days** é mutuamente exclusivo com o argumento **-to**.

Nota:

1. Certifique-se de executar o comando **planman** a partir do comando **JnextPlan**.
2. O formato utilizado para a data depende do valor designado para a variável ***date format*** especificada no arquivo `localopts`.
3. Quando um plano de produção é estendido, os números associados a prompts já presentes no plano são modificados.

Se nenhum argumento **-to**, **-for** ou **-days** for especificado, o plano de produção será estendido por um dia.

Recuperando informações sobre o plano de produção

A sintaxe a seguir é utilizada para mostrar as informações sobre o plano de produção atual:

planman [*connection_parameters*] **showinfo**

em que:

connection_parameters
: Define as configurações a serem utilizadas ao estabelecer a conexão utilizando HTTP ou HTTPS através do WebSphere Application Server com o gerenciador de domínio principal. Para obter mais informações, consulte "Linha de Comandos Planman" na página 82.

Nota: Você pode instalar o recurso Cliente da Linha de Comando do Tivoli Workload Scheduler nos agentes tolerantes a falhas e sistemas fora da rede do Tivoli Workload Scheduler para emitir, a partir desses sistemas, o comando **planman showinfo**.

A saída deste comando mostra:

- O caminho da instalação.
- O horário de início do plano de produção.
- O horário de encerramento do plano de produção.
- A duração do plano de produção, após a última extensão do plano, se estendido.
- A data e hora da última atualização do plano, feita utilizando**JnextPlan** ou **planman**.
- O horário de encerramento do plano de pré-produção.
- O horário de início da primeira instância de fluxo de tarefas não concluída.
- O *número de execuções*, que é o número total de vezes que o plano foi gerado.
- O *número de execuções confirmadas*, que é o número o vezes que o plano foi gerado com êxito.

Os horários de início e de encerramento dos planos de produção e pré-produção são exibidos utilizando o formato especificado na variável ***date format*** configurada no arquivo `localopts` e o fuso horário da máquina local.

Uma saída de amostra desse comando é a seguinte:

```
# planman showinfo
Tivoli Workload Scheduler (UNIX)/PLANMAN 8.6 (20100715)
Licensed Materials - Property of IBM*
5698-WSH
(C) Copyright IBM Corp. 1998, 2011 All rights reserved.
* Trademark of International Business Machines
Installed for user "aix61usr".
Locale LANG set to the following: "en"
Plan creation start time: 07/21/2010 06:00 TZ Europe/Rome
Production plan start time of last extension: 07/21/2010 06:00 TZ Europe/Rome
Production plan end time: 07/22/2010 05:59 TZ Europe/Rome
Production plan time extension: 024:00
```

```
Plan last update: 07/21/2010 10:05 TZ Europe/Rome
Preproduction plan end time: 08/05/2010 06:00 TZ Europe/Rome
Start time of first not complete preproduction plan job stream instance:
   07/21/2010 10:30 TZ Europe/Rome
Run number: 1
Confirm run number: 1
```

Criando um Plano Trial

A sintaxe a seguir é utilizada para criar um plano trial:

planman [*connection_parameters*] **crttrial** *file_name*

[**-from** *mm/dd/*[*yy*]*yy* [*hh[:]mm* [**tz** | **timezone** *tzname*]]]

{**-to** *mm/dd/[yy]yy[hh[:]mm[**tz** | **timezone** tzname]]* |

-for *[h]hh[:]mm* [**-days** *n*] |

-days *n*}

em que:

connection_parameters
Define as configurações a serem utilizadas ao estabelecer a conexão utilizando HTTP ou HTTPS através do WebSphere Application Server com o gerenciador de domínio principal. Para obter mais informações, consulte "Linha de Comandos Planman" na página 82.

file_name
Designa um nome para o arquivo a ser criado sob o diretório *TWS_home*/schedTrial e que contém o plano trial. O nome do arquivo que contém o plano trial é **T*filename***. Isso significa que o valor designado para *file_name* é myfile, então, o nome do arquivo que contém o plano trial gerado é Tmyfile.

-from Configura a hora de início do plano trial.

Se o argumento **-from** for omitido, então:

- A data padrão é *today*.
- A hora padrão é o valor configurado na opção global ***startOfDay*** utilizando**optman** no gerenciador de domínio principal.

-to Configura o horário de encerramento do plano trial. O argumento **-to** é mutuamente exclusivo com os argumentos **-for** and **-days**.

-for Configura a duração do plano de experiência. O formato é *hhhmm*, em que *hhh* são as horas e *mm* são os minutos. O argumento **-for** é mutuamente exclusivo com o argumento **-to**.

-days *n*
Configura o número de dias pelo qual você deseja que o plano de experiência dure. O argumento **-days** é mutuamente exclusivo com o argumento **-to**.

Nota: O formato utilizado para a data depende do valor designado para a variável ***date format*** especificada no arquivo localopts.

Se nenhum argumento **-to**, **-for** ou **-days** for especificado, a duração do plano de experiência padrão será de um dia.

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

oGuia do Usuário do Console do Dynamic Workload, seção Gerando Planos de Experiência e de Previsão.

Criando um Plano Trial de uma Extensão de Plano de Produção

A sintaxe a seguir é utilizada para criar um plano trial com a extensão do plano de produção atual:

planman [*connection_parameters*] **exttrial** *file_name*

{**-to** *mm/dd/[yy]yy[hh[:]mm[**tz** | **timezone** tzname]]* |

-for *[h]hh[:]mm* [**-days** *n*] |

-days *n*}

em que:

connection_parameters
: Define as configurações a serem utilizadas ao estabelecer a conexão utilizando HTTP ou HTTPS através do WebSphere Application Server com o gerenciador de domínio principal. Para obter mais informações, consulte "Linha de Comandos Planman" na página 82.

file_name
: Designa um nome para o arquivo a ser criado sob o diretório *TWS_home*/schedTrial e que contém o plano trial. O nome do arquivo que contém o plano trial é **T*filename***. Isso significa que o valor designado para *file_name* é myfile, então, o nome do arquivo que contém o plano trial gerado é Tmyfile.

-to
: Configura o horário de encerramento do plano de experiência que contém a extensão do plano de produção. O argumento **-to** é mutuamente exclusivo com os argumentos **-for** and **-days**.

-for
: Configura a duração do plano trial que contém a extensão do plano de produção. O formato é *hhhmm*, em que *hhh* são as horas e *mm* são os minutos. O argumento **-for** é mutuamente exclusivo com o argumento **-to**.

-days *n*
: Configura o número de dias pelo qual você deseja que o plano de experiência que contém a extensão do plano de produção dure. O argumento **-days** é mutuamente exclusivo com o argumento **-to**.

Nota: O formato utilizado para a data depende do valor designado para a variável ***date format*** especificada no arquivo localopts.

Se nenhum argumento **-to**, **-for** ou **-days** for especificado, a extensão do plano de produção padrão contida no plano de experiência será de um dia.

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

oGuia do Usuário do Console do Dynamic Workload, seção Gerando Planos de Experiência e de Previsão.

Criando um Plano de Previsão

A sintaxe a seguir é utilizada para criar um plano de previsão:

planman [*connection_parameters*] **crtfc** *file_name*

[**-from** *mm/dd/*[*yy*]*yy* [*hh*[*:*]*mm* [**tz** | **timezone** *tzname*]]]

{**-to** *mm/dd/*[*yy*]*yy*[*hh*[*:*]*mm*[***tz*** | ***timezone*** *tzname]]* |

-for *[h]hh[:]mm* [**-days** *n*] |

-**days** *n*}

em que:

connection_parameters
Define as configurações a serem utilizadas ao estabelecer a conexão utilizando HTTP ou HTTPS através do WebSphere Application Server com o gerenciador de domínio principal. Para obter mais informações, consulte "Linha de Comandos Planman" na página 82.

file_name
Designa um nome para o arquivo a ser criado sob o diretório *TWS_home*/schedForecast e que contém o plano de previsão. O nome do arquivo que contém o plano de previsão é **F*filename***. Isso significa que o valor designado para *file_name* é myfile, então, o nome do arquivo que contém o plano de previsão gerado é Fmyfile.

O comprimento máximo de *file_name* pode ter 148 caracteres.

-from Configura a hora de início do plano de previsão.

Se o argumento **-from** for omitido, então:

- A data padrão é *today*.
- A hora padrão é o valor configurado na opção global ***startOfDay*** utilizando**optman** no gerenciador de domínio principal.

-to Configura o horário de encerramento do plano previsão. O argumento **-to** é mutuamente exclusivo com os argumentos **-for** and **-days**.

-for Configura a duração do plano de previsão. O formato é *hhhmm*, em que *hhh* são as horas e *mm* são os minutos. O argumento **-for** é mutuamente exclusivo com o argumento **-to**.

-days *n*
Configura o número de dias pelo qual você deseja que o plano de previsão dure. O argumento **-days** é mutuamente exclusivo com o argumento **-to**.

Nota: O formato utilizado para a data depende do valor designado para a variável ***date format*** especificada no arquivo localopts.

Se nenhum argumento **-to**, **-for** ou **-days** for especificado, a duração do plano de previsão padrão será de um dia.

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

oGuia do Usuário do Console do Dynamic Workload, seção Gerando Planos de Experiência e de Previsão.

Regras de Implementação

O comando `planman deploy` é utilizado no gerenciamento de eventos. É possível utilizá-lo para implementar manualmente todas as regras que não estão no estado de rascunho (a propriedade `isDraft` é configurada como `NO` em suas definições). O comando opera da seguinte maneira:

1. Seleciona todas as definições de regras de eventos que não estão no estado de rascunho a partir do banco de dados do Tivoli Workload Scheduler.
2. Constrói arquivos de configuração de regra de evento.
3. Implementa os arquivos de configuração nos mecanismos de monitoramento em execução nos agentes do Tivoli Workload Scheduler.

Os novos arquivos de configuração atualizam as regras de eventos em execução em cada mecanismo de monitoramento em termos de:
- Novas regras
- Regras alteradas
- Regras excluídas ou configuradas de volta para o estado`draft`

É possível utilizar este comando além da, ou em substituição à, opção de configuração `deploymentFrequency (df) optman`, que verifica periodicamente definições de regra de evento para alterações na implementação (consulte o *Guia de Administração* para obter detalhes sobre esta opção).

As alterações aplicadas às definições de regras de eventos no banco de dados são efetivadas somente depois que a implementação tiver ocorrido.

A sintaxe de comando é:

planman [*connection_parameters*] **deploy** [**-scratch**]

em que:

connection_parameters
: Define as configurações a serem utilizadas ao estabelecer a conexão utilizando HTTP ou HTTPS através do WebSphere Application Server com o gerenciador de domínio principal. Para obter mais informações, consulte "Linha de Comandos Planman" na página 82.

-scratch
: Sem esta opção, o comando afeta apenas as regras que foram incluídas, alteradas, excluídas ou configuradas de volta para o estado`rascunho`.

  Com esta opção, o comando implementa todas as regras de não rascunho existentes no banco de dados, incluindo aquelas que já estão na implementação e não foram alteradas.

Observe que o tempo de implementação aumenta proporcionalmente com o número de regras ativas para implementação. Se for necessário implementar um grande número de regras novas ou alteradas, execute `planman deploy` com esta opção para reduzir o tempo de implementação.

A utilização desta opção resulta em uma reconfiguração completa do processador de eventos e deve ser utilizada com cuidado. O comando pode causar a perda de quaisquer instâncias da regra em progresso no momento em que você o emite. O caso típico é uma regra seqüencial que foi acionada e que está aguardando que eventos adicionais ocorram: se você utilizar a opção neste momento, o ambiente da regra de evento será reconfigurado e os eventos rastreados serão perdidos.

Para executar este comando, é necessário o acesso **build** no arquivo **prodsked**.

Desbloqueando o Plano de Produção

Quando oTivoli Workload Scheduler começa a criar o plano de produção, ele bloqueia as definições dos objetos de planejamento no banco de dados e, em seguida, desbloqueia as mesmas quando a criação do plano de produção é concluída ou se ocorrer uma condição de erro. O bloqueio é aplicado para evitar que definições de objetos sejam modificadas quando o plano de produção é gerado ou estendido. Se o processamento terminar de formal inesperada, as entradas do banco de dados podem permanecer bloqueadas. Somente usuários com acesso **build** no tipo de objeto do arquivo **prodsked** especificados no arquivo de segurança no gerenciador de domínio principal têm permissão para desbloquear o banco de dados. O comando utilizado para executar essa ação é:

planman [*connection_parameters*] **unlock**

em que:

connection_parameters
Define as configurações a serem utilizadas ao estabelecer a conexão utilizando HTTP ou HTTPS através do WebSphere Application Server com o gerenciador de domínio principal. Para obter mais informações, consulte "Linha de Comandos Planman" na página 82.

Nota: Você pode instalar o recurso Cliente da Linha de Comando do Tivoli Workload Scheduler nos agentes tolerantes a falhas e sistemas fora da rede do Tivoli Workload Scheduler para emitir, a partir desses sistemas, o comando **planman unlock**.

Reconfigurando o plano de produção

O script a seguir é utilizado para reconfigurar ou riscar o plano de produção:

ResetPlan [*connection_parameters*] [**-scratch**]

em que:

connection_parameters
Define as configurações a serem utilizadas ao estabelecer a conexão utilizando HTTP ou HTTPS através do WebSphere Application Server com o gerenciador de domínio principal. Para obter mais informações, consulte "Linha de Comandos Planman" na página 82.

A diferença entre reconfigurar e riscar o plano de produção é a seguinte:

- Se você *reconfigurar* o plano de produção, o plano de pré-produção é mantido, ele é atualizado com as estatísticas de tarefas e é utilizado posteriormente para gerar um novo plano de produção. Isso significa que ao criar um novo plano de produção, ele conterá todas as instâncias do fluxo de tarefas que não estavam no estado COMPLETE quando você executou **ResetPlan**. As etapas executadas pelo produto ao reconfigurar o plano de produção são as seguintes:
  1. O arquivo `Symphony` é arquivado.
  2. As estatísticas de tarefa são atualizadas.
- Se você *riscar* o plano de produção, o plano de pré-produção também é riscado. O plano de pré-produção será criado novamente com base nas informações de modelagem armazenadas no banco de dados quando você gerar posteriormente um novo plano de produção. Isso significa que o novo plano de produção conterá todas as instâncias do fluxo de tarefas planejadas para execução no espaço de tempo coberto pelo plano, independentemente do fato de já estarem no estado COMPLETE ou não quando o plano foi riscado. As etapas executadas pelo produto ao riscar o plano de produção são as seguintes:
  1. O arquivo `Symphony` atual é arquivado e os dados do plano replicados no banco de dados são excluídos.
  2. As estatísticas de tarefa são atualizadas.
  3. O plano de pré-produção é riscado.

  Nota: Se você utilizar a opção **-scratch**, certifique-se de executar **dbrunstats** antes do script **JnextPlan**. Consulte o *Guia de Administração* para obter detalhes sobre `dbrunstats`.

Ao executar o comando **`ResetPlan`** um arquivo de log da tarefa é criado no diretório <*TWS_INST_DIR*>\TWS\stdlist\<*DATE*>, em que <*TWS_INST_DIR*> é o diretório de instalação do Tivoli Workload Scheduler e <*DATE*> é a data em que o script foi executado.

Removendo o Plano de Pré-Produção

O seguinte script é usado para remover o plano de pré-produção enquanto mantém o arquivo `Symphony`:

Planman reset -scratch

Ao executar esse comando, o plano de pré-produção é riscado. O plano de pré-produção será criado novamente com base nas informações de modelagem armazenadas no banco de dados quando você gerar posteriormente um novo plano de produção. Isso significa que o novo plano de produção conterá todas as instâncias do fluxo de tarefas planejadas para execução no espaço de tempo coberto pelo plano, independentemente do fato de já estarem no estado COMPLETE ou não quando o plano foi riscado. As etapas executadas pelo produto ao riscar o plano de produção são as seguintes:

1. Todos os processos do Tivoli Workload Scheduler são parados no gerenciador de domínio principal.
2. O arquivo `Symphony` é mantido.
3. As estatísticas de tarefa são atualizadas.
4. O plano de pré-produção é riscado.

Nota: Se você utilizar a opção **-scratch**, certifique-se de executar **dbrunstats** antes do script **JnextPlan**. Consulte o *Guia de Administração* para obter detalhes sobre `dbrunstats`.

Replicando os Dados do Plano no Banco de Dados

Realinha os dados do plano no banco de dados com os dados no arquivo Symphony.

O armazenamento de informações sobre objetos no plano em um banco de dados torna o acesso aos dados do plano mais fácil e mais rápido. Quando grandes números de usuários estiverem acessando simultaneamente o ambiente de backend do Tivoli Workload Scheduler, o desempenho e a confiabilidade poderão ser comprometidos. Ao replicar o plano em um banco de dados relacional, os usuários podem acessar os dados de maneira rápida e confiável.

Para versões anteriores à 9.1, as seguintes operações requerem acesso ao plano do `Symphony`:

- Executando relatórios de linha de base
- Exibindo o plano em uma visualização gráfica
- Exibindo uma visualização de impacto
- Acionando ações
- Atualizando visualizações de monitoramento de tarefas e de fluxos de tarefas

Todas essas atividades de modelagem ou monitoramento requeriam acesso ao plano e, se você multiplicar esses cenários pelo número de usuários que solicitam acesso simultâneo ao plano para executar uma ou mais dessas atividades, o resultado será tempos de resposta e de desempenho geral lentos.

Para resolver esse problema, o Tivoli Workload Scheduler replica o plano em um banco de dados relacional em que as instruções SQL são usadas para recuperar dados de forma rápida e confiável. Uma caixa de mensagens nova, `mirrorbox.msg`, é usada para sincronizar o banco de dados com o arquivo Symphony. Se o `mirrorbox.msg` ficar cheio, por exemplo, se o banco de dados permanecer indisponível por um longo período de tempo, o plano será recarregado automaticamente no banco de dados usando as informações no arquivo Symphony.

Em particular, o desempenho e o tempo de resposta do script UpdateStats foram melhorados consideravelmente com essa nova forma de gerenciar dados do plano.

Além disso, cada vez que o plano for estendido, o plano no banco de dados será eliminado e recriado, disponibilizando as informações mais recentes para todos os usuários. Para replicar dados do plano manualmente a partir do arquivo Symphony para o banco de dados, execute o comando **`planman resync`**.

A sincronização do arquivo Symphony com o banco de dados é ativada automaticamente ao incluir o arquivo Sfinal no banco de dados com o comando composer **`add Sfinal`**. Uma nova tarefa, CHECKSYNC, foi incluída no fluxo de tarefas FINALPOSTREPORTS contido no arquivo Sfinal, que é responsável por monitorar o estado do processo de replicação do arquivo Symphony no banco de dados. Ele envia o progresso e o status desse processo para o log da tarefa. Se a tarefa CHECKSYNC falhar, consulte o log da tarefa CHECKSYNC, bem como o log do WebSphere Application Server para determinar o problema. Após resolver o problema, execute o comando **`planman resync`** para recarregar os dados do plano do arquivo Symphony no banco de dados.

Importante: Se você estiver fazendo upgrade do ambiente do Tivoli Workload Scheduler para a versão 9.2, em seguida, haverá algumas etapas manuais que você precisa implementar antes que os dados do plano sejam replicados no banco de

dados. Consulte Customizando e Enviando o Fluxo de Tarefas Final Opcional e “Automatizando o processamento do plano de produção” na página 101 para obter mais informações.

Para simplificar as integrações, é fornecido um conjunto de visualizações do banco de dados para um conjunto de tabelas que contém dados do plano no banco de dados do Tivoli Workload Scheduler.

Para ver as visualizações do banco de dados que contêm informações sobre objetos do Tivoli Workload Scheduler no plano, consulte a visualizações que iniciam com "PLAN_" em *Tivoli Workload Scheduler: Visualizações do Banco de Dados*.

Ao executar operações a partir do Dynamic Workload Console que recuperam os dados do plano atual, se suspeitar que os dados não estão atualizados, será possível executar **planman resync** para atualizar os dados do plano no banco de dados com as informações mais recentes no arquivo Symphony. Se a caixa de mensagens, mirrorbox.msg, responsável por sincronizar o banco de dados com o arquivo Symphony ficar cheia, por exemplo, o banco de dados está indisponível por um longo período de tempo, um **planman resync** será emitido automaticamente para que o plano seja totalmente recarregado no banco de dados.

A sintaxe a seguir é usada para replicar os dados do plano no banco de dados com os dados no arquivo do Symphony:

planman [*connection_parameters*] **resync**

em que:

connection_parameters
: Define as configurações a serem utilizadas ao estabelecer a conexão utilizando HTTP ou HTTPS através do WebSphere Application Server com o gerenciador de domínio principal. Para obter mais informações, consulte “Linha de Comandos Planman” na página 82.

Monitorando a Replicação dos Dados do Plano no Banco de Dados

Monitora o progresso e o resultado do processo de replicação do arquivo do Symphony no banco de dados.

Monitora o progresso e o resultado do processo de replicação do arquivo do Symphony no banco de dados. As mensagens são gravadas na saída padrão com o progresso e status do comando. O comando `planman checksync` também é definido na tarefa, `CHECKSYNC`, que está contida no fluxo de tarefas FINALPOSTREPORTS incluído no arquivo Sfinal. Se a tarefa `CHECKSYNC` falhar, consulte o log da tarefa da tarefa `CHECKSYNC`, bem como o log do WebSphere Application Server para determinar o problema. Após resolver o problema, execute o comando `planman resync` para recarregar os dados do plano do arquivo Symphony no banco de dados.

A sintaxe a seguir é usada para monitorar o progresso e o resultado da replicação dos dados do plano no banco de dados com os dados no arquivo do Symphony:

planman [*connection_parameters*] **checksync**

em que:

connection_parameters
Define as configurações a serem utilizadas ao estabelecer a conexão utilizando HTTP ou HTTPS através do WebSphere Application Server com o gerenciador de domínio principal. Para obter mais informações, consulte "Linha de Comandos Planman" na página 82.

O comando stageman

O comando **stageman** transporta o saldo de fluxos de tarefa incompletos, arquiva o plano de produção antigo e instala o novo plano de produção. Uma cópia de `Symphony` é enviada para os gerenciadores de domínio e agentes como parte do processo de inicialização do novo plano de produção. Ao executar **JnextPlan**, **stageman** é chamado a partir do script **SwitchPlan**.

Você deve ter acesso ***build*** para o arquivo `Symphony` para executar **stageman**.

Sintaxe

stageman -V | -U

stageman
[**-carryforward**{**yes** | **no** | **all**}]
[**-log** *log_file* | **-nolog**]
[*symnew*]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

-carryforward
Define como os fluxos de tarefas incompletos são gerenciados ao mover para um novo plano de produção. As configurações disponíveis são:

não Não transporta o saldo de nenhum fluxo de tarefa.

sim Transporta o saldo somente dos fluxos de tarefas incompletos cuja definição contém a palavra-chave **carryforward**.

all Transporta o saldo de todos os fluxos de tarefas incompletos, independentemente de conterem ou não a palavra-chave **carryforward** na definição do fluxo de tarefas.

Se você omitir essa palavra-chave, por padrão, ela será configurada para o valor especificado globalmente utilizando **optman** para a opção *enCarryForward* . Consulte "Entendendo as Opções de Transporte de Saldo" na página 71 para obter informações sobre a configuração de transporte de saldo quando a opção global *enCarryForward* e as palavras-chave **-carryforward** forem configuradas.

-log Arquiva o plano de produção antigo no diretório *TWS_home*`/schedlog` com o nome de arquivo *log_file*. Os planos de produção arquivados poderão, então, ser listados e selecionados utilizando os comandos "listsym" na página 388 e "setsym" na página 400. Se as palavras-chave **-log** e **-nolog** não forem especificadas, o Tivoli Workload Scheduler arquiva o plano de produção antigo, utilizando a seguinte convenção de nomenclatura:

`Myyyymmddhhtt`

em que *yyyymmddhhtt* corresponde ao ano, mês, dia, hora, minutos em que o plano de produção antigo está arquivado. Se você gerar o plano de produção utilizando **JnextPlan**, poderá customizar essa convenção de nomenclatura no script **SwitchPlan**.

Nota: Certifique-se de monitorar o espaço em disco no diretório `schedlog` e remover os arquivos de log mais antigos regularmente.

-nolog Não arquiva o antigo plano de produção.

symnew
O nome designado para o arquivo do plano de produção intermediário criado por **planman**. Se não especificado, **stageman** utiliza o nome de arquivo `Symnew`.

Comentários

Para permitir que os procedimentos de transporte de saldo funcionem corretamente em uma rede, o arquivo do plano de produção, `Symphony`, do gerenciador de domínio principal deve ser atualizado com o status mais recente do fluxo de tarefas de seus agentes e gerenciadores de domínio subordinados. Execute o seguinte comando:

```
conman "link @"
```

antes de executar **stageman**. Esse comando vincula quaisquer estações de trabalho não vinculadas de forma que mensagens sobre o status de processamento da tarefa enfileiradas no arquivo `Mailbox.msg` sejam enviadas de volta ao gerenciador de domínio principal para atualizar o arquivo `Symphony`.

Nota: Somente no UNIX, **stageman** determina também quais arquivos executáveis associados a tarefas enviadas utilizando os comandos utilitários **at** e **batch** podem ser excluídos quando o Tivoli Workload Scheduler for iniciado para o novo período de produção. Essas tarefas não têm o saldo transportado.

Exemplos

Transporte todos os fluxos de tarefas não concluídos (independentemente do status da opção Carry Forward), registre o arquivo `Symphony` antigo e crie o novo arquivo `Symphony`:

```
DATE='datecalc today pic YYYYMMDDHHTT'
stageman -carryforward all -log schedlog/M$DATE
```

Transporte o saldo dos fluxos de tarefa não concluídos conforme definido pela opção global ***carryforward***, não registre o arquivo `Symphony` antigo e crie um plano de produção intermediário denominado `mysym`:

```
stageman -nolog mysym
```

Gerenciando Acessos Simultâneos ao Arquivo Symphony

Esta seção contém dois cenários de amostra que descrevem como o Tivoli Workload Scheduler gerencia possíveis acessos simultâneos ao arquivo `Symphony` ao executar **stageman**.

Cenário 1: Acesso ao Arquivo Symphony Bloqueado por Outros Processos do Tivoli Workload Scheduler

Se os processos do Tivoli Workload Scheduler ainda estiverem ativos e acessando o arquivo `Symphony` quando**stageman** for executado, a mensagem a seguir é exibida:

```
Impossível obter acesso exclusivo para o Symphony.
Encerre o batchman e o mailman.
```

Para continuar, pare o Tivoli Workload Scheduler e execute o**stageman** novamente. Se o**stageman** abortar por qualquer razão, você deve executar novamente o**planman** e o**stageman**.

Cenário 2: Acesso ao Arquivo Symphony Bloqueado pelo stageman

Se você tentar acessar o plano utilizando a interface de linha de comandos enquanto o`Symphony` está sendo comutado, receberá a seguinte mensagem:

```
Arquivo Symphony atual é antigo. Mudando para novo Symphony.
Planejar mm/dd/aaaa (nnnn) na cpu, Symphony comutado.
```

Gerenciando Dependências Follows Utilizando Prompt de Transporte de Saldo

Para manter a continuidade ao transportar saldo de fluxos de tarefa, o **stageman** gera prompts para cada fluxo de tarefas que tiver o saldo transportado e tiver uma dependência `follows` em outro fluxo de tarefas que não terá o saldo transportado. Esses prompts são emitidos depois do início do novo período de processamento, quando o Tivoli Workload Scheduler verifica se a tarefa ou o fluxo de tarefas está pronto para ser ativado, e são respondidos como prompts padrão. Segue um exemplo de um *prompt para transportar saldo*:

```
INACT 1(SYS2#SKED2[(0600 01/11/06),(0AAAAAAAAAAAAA2Y)]) follows SYS1#SKED1, satisfied?
      SYS1#SKED1, satisfied?
```

Esse prompt indica que um fluxo de tarefas, que tem o saldo transportado do plano de produção anterior, `(SYS2#SKED2[(0600 01/11/06),(0AAAAAAAAAAAAA2Y)])`, tem uma dependência follows de um fluxo de tarefas denominado `SYS1#SKED1` que não teve o saldo transportado. Para obter informações sobre a sintaxe utilizada para indicar o fluxo de tarefa com saldo transportado, consulte “Selecionando Fluxos de Tarefas em Comandos” na página 353.

O estado do prompt, **INACT**, nesse caso, define o estado da dependência `follows` correspondente. Os estados possíveis são:

INACT
: O prompt não foi emitido e a dependência não é satisfeita.

PERGUNTADO
: O prompt foi emitido e está aguardando uma resposta. A dependência não é satisfeita.

NÃO Uma resposta "no" foi recebida ou determinou-se antes de transportar saldo ter ocorrido que o fluxo de tarefa seguido (`SKED3`) não tinha sido concluído com êxito. A dependência não é satisfeita.

SIM Uma resposta "yes" foi recebida ou determinou-se antes de transportar saldo ter ocorrido que o fluxo de tarefa seguido (`SKED3`) tinha sido concluído com êxito. A dependência é satisfeita.

O comando logman

O comando **logman** registra as estatísticas do job de um arquivo de log do plano de produção.

Sintaxe

logman -V | -U

logman
[*connectionParameters*]
{**-prod** | *symphony-file*]
[**-smooth** *weighting*]
[**-minmax** {*elapsed* | *cpu*}]}

Argumentos

-U Exibe informações de utilização do comando e sai.

-V Exibe a versão do comando e sai.

connectionParameters
Representa o conjunto de parâmetros que controlam a interação entre a interface do produto, **logman** em execução no gerenciador de domínio principal neste caso, e a infra-estrutura do WebSphere Application Server utilizando HTTP ou HTTPS. Utilize essa sintaxe para especificar as configurações para os parâmetros de conexão:

[**-host** *hostname*] [**-port** *port_number*] [**-protocol** *protocol_name*] [**-proxy** *proxy_name*] [**-proxyport** *proxy_port_number*] [**-password** *user_password*] [**-timeout** *timeout*] [**-username** *username*]

em que:

hostname
O nome do host do gerenciador de domínio principal.

port_number
O número da porta usado ao estabelecer a conexão com o gerenciador de domínio principal.

protocol_name
O protocolo utilizado durante a comunicação. Pode ser HTTP com autenticação básica ou HTTPS com autenticação de certificado.

proxy_name
O nome do host do proxy utilizado na conexão.

proxy_port_number
O número da porta do proxy utilizado na conexão.

user_password
A senha do usuário que é utilizada para executar o **logman**.

Nota: Em estações de trabalho Windows, quando você especificar uma senha que contém aspas duplas (") ou outros caracteres especiais, certifique-se de que o caractere seja codificado com a função escape. Por exemplo, se a sua senha for `tws11"tws`, grave-a assim: `"tws11\"tws"`.

timeout
O tempo máximo, expresso em segundos, que o programa de linha de comandos que está conectando pode aguardar pela resposta do gerenciador de domínio principal antes de considerar a solicitação de comunicação como com falha.

username
O nome do usuário que está executando o **logman**.

Se qualquer um desses parâmetros for omitido, ao chamar **logman**, o Tivoli Workload Scheduler procura um valor primeiro no arquivo `useropts` e, em seguida, no arquivo `localopts`. Se uma configuração para o parâmetro não for localizada, um erro é exibido. Consulte "Configurando Opções para Utilização das Interfaces com o Usuário" na página 53 para obter informações sobre os arquivos `useropts` e `localopts`.

-prod Atualiza o plano de pré-produção com as informações sobre os fluxos de tarefas no estado COMPLETE na produção. Fazendo isso o plano de pré-produção é mantido atualizado com as informações de processamento mais recentes. Isto evita a possibilidade do novo plano de produção ser executado novamente, fluxos de tarefas já concluídos no período de produção anterior.

-minmax {*elapsed* | *cpu*}
Define como os tempos de execução mínimo e máximo do job são registrados e relatados. As configurações disponíveis são:

elapsed
Basear os tempos de execução mínimo e máximo no tempo decorrido.

cpu Basear os tempos de execução mínimo e máximo no tempo de CPU.

Esta configuração é utilizada quando o comando **logman** é executado a partir da linha de comandos e não pelo script **JnextPlan**. Quando o comando **logman** é executado por **JnextPlan**, a configuração utilizada é aquela especificada na opção global *logmanMinMaxPolicy*.

-smooth *weighting*
Utiliza um fator de pesagem que favorece a execução do job mais recente ao calcular o tempo de execução normal (média) de um job. Isso é expresso como porcentagem. Por exemplo, **-smooth 40** aplica um fator de peso de 40% na execução da tarefa mais recente e 60% na média existente. O padrão é 0%. Esta configuração é utilizada quando o comando **logman** é executado a partir da linha de comandos e não pelo script **JnextPlan**. Quando o comando **logman** é executado pelo **JnextPlan**, a configuração utilizada é aquela especificada na opção global *logmanSmoothPolicy*.

symphony-file
O nome de um arquivo symphony arquivado a partir do qual as estatísticas de tarefa são extraídas.

Comentários

Os jobs que já foram registrados não podem ser registrados novamente. A tentativa de fazer isso gera uma mensagem de erro `0 jobs logged`.

Exemplos

Registrar estatísticas do job do arquivo de log M199903170935:

```
logman schedlog/M199903170935
```

Como a Média dos Tempos de Execução É Calculada

A duração estimada de uma execução da tarefa é fornecida por logman como parte do ciclo de planejamento diário. A duração estimada de uma execução da tarefa é baseada na média de suas execuções precedentes. Para calcular o tempo de execução médio para uma tarefa, logman divide o tempo de execução total para todas as execuções bem-sucedidas pelo número de execuções bem-sucedidas. Se um grande número de execuções for utilizado para calcular a média, uma alteração repentina em um tempo de execução da tarefa não será refletida imediatamente na média. Para responder mais rapidamente a tais alterações, é possível utilizar a opção `smooth` para que a média possa ser ponderada em favor das execuções da tarefa mais recentes. Utilize a opção `-smooth` para inserir um fator de ponderação, como uma porcentagem, para execuções da tarefa atuais. Por exemplo, o comando `logman -smooth 40` fará com que logman utilize um fator de ponderação de 40 por cento para as execuções mais recentes da tarefa e 60 por cento para a média existente. O comando `logman -smooth 100` fará com que as execuções mais recentes da tarefa substituam a média existente.

Logman retém os dados de estatística de execuções de tarefa no banco de dados do Tivoli Workload Scheduler. Não há limite para o número de instâncias da tarefa retidas no histórico da tarefa.

Iniciando o Processamento do Plano de Produção

Para iniciar um ciclo de produção, siga estas etapas:

1. Efetue login como *TWS_user* no gerenciador de domínio principal.
2. Em um prompt de comandos, execute o comando script **. ./***TWS_home*/**tws_env.sh** no UNIX ou *TWS_home***tws_env.cmd** no Windows para configurar o ambiente e, em seguida, execute a tarefa `JnextPlan`, digitando, por exemplo, em uma estação de trabalhoUNIX, o seguinte comando:

   ```
   JnextPlan.sh -from 05/03/06 0400 -to 06/06/06
   ```

   Isso cria um novo plano de produção que começa em 3 de maio de 2006 às 4h e pára em 6 de junho de 2006 às 3h59. O dia de processamento será iniciado na hora especificada no gerenciador de domínio principal na variável ***startOfDay***.
3. Quando o job `JnextPlan` for concluído, verifique o status do Tivoli Workload Scheduler:

   ```
   conman status
   ```

   Se oTivoli Workload Scheduler foi iniciado corretamente, o status será

   ```
   Batchman=LIVES
   ```

   Se o mailman ainda estiver executando um processo na estação de trabalho remota, você verá que ela não é inicializada imediatamente. Isso acontece porque a estação de trabalho precisa concluir quaisquer atividades contínuas que envolvem o processo mailman antes de reinicializar. Depois do intervalo definido no parâmetro***mm retry link*** configurado no arquivo de configuração *TWS_home*/`localopts` decorrer, o gerenciador de domínio tenta inicializar a estação de trabalho novamente. Assim que as atividades contínuas são concluídas, as atividades do dia seguinte são inicializadas. Para obter informações sobre o arquivo de configuração `localopts`, consulte o *Guia de Administração do IBM Tivoli Workload Scheduler*.
4. Aumente o limite para permitir a execução das tarefas. O limite de job padrão após a instalação é zero. Isso significa que nenhum job será executado.

   ```
   conman "limit;10"
   ```

Automatizando o processamento do plano de produção

Se você desejar estender seu plano de produção em um intervalo de tempo fixado, por exemplo, a cada semana, você terá a opção de automatizar a extensão. Esta seção explica como fazer isso.

Com o Tivoli Workload Scheduler versão 9.1 e posterior, o arquivo Sfinal foi modificado para incluir dois fluxos de tarefas de amostra nomeados `FINAL` e `FINALPOSTREPORTS` que ajudam a automatizar o gerenciamento de planos. Uma cópia destes fluxos de tarefas é localizada no arquivo `Sfinal` no diretório *`TWS_home`*. Além disso, uma cópia dos scripts da tarefa também está localizada nesse mesmo local.

É possível utilizar esse arquivo `Sfinal` ou criar e customizar um novo.

Importante: Em qualquer caso, para poder executar estes fluxos de tarefa com sucesso, o *TWS_user* deve ter acesso `Write` ao diretório temporário padrão `tmp`.

O fluxo de tarefas `FINAL` executa a sequência de arquivos de script descritos em `JnextPlan` para gerar o novo plano de produção. Consulte “Criando e estendendo o plano de produção” na página 79 para obter referência.

O fluxo de tarefas `FINALPOSTREPORTS`, responsável por imprimir relatórios pós-produção, segue o fluxo de tarefas `FINAL` e é iniciado apenas quando a última tarefa listada no fluxo de tarefas `FINAL` (SWITCHPLAN) tiver sido concluída com êxito. O fluxo de tarefas FINALPOSTREPORTS também inclui uma tarefa nomeada, CHECKSYNC, que monitora o progresso e o resultado do comando **`planman resync`**. O comando **`planman resync`** carrega os dados do plano a partir do arquivo Symphony para o banco de dados.

Com o Tivoli Workload Scheduler versão 9.1 e posterior, os dados do plano agora são totalmente replicados no banco de dados. Cada vez que o plano for estendido, o plano no banco de dados será recriado, disponibilizando as informações mais recentes para todos os usuários. Se você estiver executando uma nova instalação do Tivoli Workload Scheduler, em seguida, essa sincronização do arquivo Symphony com o banco de dados será ativada automaticamente ao incluir o arquivo Sfinal no banco de dados com o comando composer **`add Sfinal`**. Como o arquivo Sfinal não é sobrescrito durante um upgrade, você deverá incluir os fluxos de tarefas FINAL e FINALPOSTREPORTS atualizados no banco de dados executando o comando composesr add Sfinal. Isso assegura que você tenha a tarefa CHECKSYNC que é responsável por replicar os dados do plano no banco de dados. O Sfinal atualizado final pode ser localizado em `TWA_home/config/diretório`. Em seguida, você deve executar JnextPlan para incluir os fluxos de tarefas FINAL e FINALPOSTREPORTS no plano de produção atual. Consulte o Customizando e Enviando o Fluxo de Tarefas Final Opcional.

Por padrão, o fluxo de tarefas `FINAL` é configurado para ser executado uma vez por dia, seguido pelo fluxo de tarefas `FINALPOSTREPORTS`. É possível modificar o horário de execução dos fluxos de tarefas, modificando duas configurações na definição de fluxo de tarefas. Esses são os detalhes sobre as duas etapas que precisam ser seguidas para fazer isso, por exemplo, para que os fluxos de tarefas sejam executados a cada três dias:

- Planeje o fluxo de tarefas para ser executado a cada três dias, modificando o ciclo de execução dentro da definição de fluxo de tarefas.

- Na instrução que chama **MakePlan** dentro do fluxo de tarefas FINAL, configure o plano de produção para durar três dias, especificando **-for 72**.

Em seguida, é necessário incluir os fluxos de tarefas no banco de dados, executando as seguintes etapas:

1. Efetue login como *TWS_user*.
2. Execute o script **tws_env** para definir o ambiente do Tivoli Workload Scheduler como a seguir:
   - UNIX: on C shells launch `. ./TWS_home/tws_env.csh`
   - UNIX: em shells Korn, ative `. ./TWS_home/tws_env.sh`
   - Em uma linha de comandos do Windows: ative `TWS_home\tws_env.cmd`

   em que *TWS_home* representa o diretório de instalação do produto.
3. Inclua as definições de fluxo de tarefa FINAL e FINALPOSTREPORTS no banco de dados executando o comando a seguir:

   ```
   composer add Sfinal
   ```

   Se você não utilizou o arquivo Sfinal fornecido com o produto, mas criou um novo, utilize seu nome no lugar de Sfinal.
4. Inicie o ciclo de produção executando o script **JnextPlan**. Dessa forma, os fluxos de tarefas FINAL e FINALPOSTREPORTS serão incluídos no plano de produção atual.

Nota: Mesmo se tiver decidido automatizar a extensão do plano de produção, você ainda pode executar **JnextPlan** a qualquer momento.

Capítulo 5. Utilizando Garantia do Serviço de Carga de Trabalho

A garantia de serviço de carga de trabalho é um recurso opcional que fornece um meio de sinalizar tarefas como essenciais para seu negócio e garantir que elas sejam processadas em tempo hábil. O uso desta função beneficia sua equipe de operações de planejamento aprimorando sua capacidade de atender níveis de serviço definidos.

Quando o recurso de garantia de serviço de carga de trabalho está ativado, é possível sinalizar tarefas como críticas para a missão e garantir que elas tenham um prazo final de conclusão associado especificado em suas definições ou no envio. Dois encadeamentos adicionais de execução, Planejador de Tempo e Monitor do Plano, que são executados no WebSphere Application Server, são, posteriormente, incluídos para garantir que as tarefas críticas sejam concluídas no prazo.

A definição de uma tarefa crítica e de um prazo final aciona o cálculo dos horários de início de todas as outras tarefas que são predecessoras da tarefa crítica. O conjunto de predecessoras de uma tarefa crítica formam sua *rede crítica*. Isto pode incluir tarefas de outros fluxos de tarefas. Começando pelo prazo final e duração da tarefa crítica, o Planejador de Tempo calcula seu *horário de início crítico*, que é o horário de início mais recente para a tarefa se adaptar ao prazo final correspondente. Retornando do horário de início crítico, ele calcula o último horário no qual cada predecessor na rede crítica pode iniciar para que a tarefa crítica no final da cadeia possa concluir no prazo.

Enquanto o plano é executado, o Monitor de Planos verifica constantemente a rede crítica para garantir que o prazo final da tarefa crítica possa ser atendido. Quando alterações que têm um impacto no tempo são feitas na rede crítica, por exemplo, a inclusão ou a remoção de tarefas ou dependências que as seguem, o Monitor de Planos solicita que o Planejador de Tempo recalcule os horários de início críticos. Além disso, quando uma tarefa da rede crítica é concluída, o tempos das tarefas que a seguem são recalculados para levar em consideração a duração real da tarefa.

Em uma rede crítica, o conjunto de predecessores que arriscam mais diretamente a atrasar o horário de início crítico é chamado de *caminho crítico*. O caminho crítico é dinamicamente atualizado à medida que as predecessoras são concluídas ou que se arriscam a concluir as alterações atrasadas.

O planejador (batchman) age automaticamente para remediar atrasos, priorizando tarefas que estão realmente ou possivelmente colocando o prazo final de destino em risco. Uma série de visualizações de tarefas críticas especializadas, disponíveis no Dynamic Workload Console, permite que os operadores procurem tarefas crítica, exibam seus predecessores e os caminhos críticos associados a eles, identifiquem tarefas que estão causando problemas e façam pesquisas detalhadas para identificar e resolver problemas.

Para obter informações detalhadas, consulte:

- "Ativando e Configurando a Workload Service Assurance" na página 104
- "Planejando Tarefas Críticas" na página 106

© Copyright IBM Corp. 1999, 2014

- "Processando e Monitorando Tarefas Críticas" na página 108
- "Cenário de Garantia de Serviço de Carga de Trabalho" na página 110

Para obter informações sobre resolução de problemas e problemas comuns com a garantia do serviço da carga de trabalho, consulte o capítulo Garantia de Serviço de Carga de Trabalho no *Tivoli Workload Scheduler: Resolução de Problemas*.

Ativando e Configurando a Workload Service Assurance

Uma grande quantidade de opções globais e locais controlam o gerenciamento de tarefas críticas. O arquivo de segurança do Tivoli Workload Scheduler também deve autorizar usuários com acesso adequado a todas as tarefas, fluxos de tarefas e estações de trabalho associados às tarefas críticas.

Opções Globais

O recurso de garantir do serviço de carga de trabalho é ativado e desativado pela opção global `enWorkloadServiceAssurance`. Ele é ativado por padrão. Outras opções globais e locais são utilizadas para controlar diferentes aspectos do processamento de tarefas críticas e suas predecessoras.

A Tabela 14 mostra as opções globais que são utilizadas pela garantia do serviço de carga de trabalho. Se você desejar customizar os valores, modifique as opções globais no gerenciador de domínio principal utilizando a linha de comandos **optman**. Na maioria dos casos, as alterações tomam efeito depois que a próxima JnextPlan é executada.

Tabela 14. Opções Globais de Garantia de Serviço de Carga de Trabalho

Opção	Descrição
enWorkloadServiceAssurance | wa	Ativa ou desativa o processamento privilegiado de tarefas críticas para a missão e seus predecessores. O valor padrão é YES. Especifique NO para desativar.
promotionOffset | po	A garantia do serviço de carga de trabalho calcula um horário de início crítico para a própria tarefa crítica e cada uma de suas predecessoras. Esse é o último horário que a tarefa pode ser iniciada sem colocar a conclusão atual da tarefa crítica em risco. Quando a hora de início crítica de uma tarefa está se aproximando e a tarefa ainda não começou, o mecanismo de promoção é utilizado. Uma tarefa promovida recebe recursos de sistema operacional adicionais e seu envio tem prioridade. A opção global `promotionoffset` determina a duração do tempo antes do horário de início crítico que uma tarefa se torna elegível para promoção. A configuração padrão é 120 segundos.
longDurationThreshold | ld	O cálculo de horários de início críticos para as tarefas que formam uma rede crítica é baseado no prazo final definido para a tarefa crítica e nas durações estimadas da tarefa crítica e de cada uma de suas predecessoras. Se uma tarefa levar mais tempo do que o esperado para que seja concluída, ela poderá fazer com que as tarefas que vêm em seguida percam seus horários de início críticos e, dessa forma, coloque a conclusão adequada da tarefa crítica em risco. A opção global `longDurationThreshold` é um valor em porcentagem. O padrão é 150. Utilizando o valor padrão, se a duração real de uma tarefa for superior a 150% da duração estimada ou mais, a tarefa será considerada uma tarefa de longa duração e será incluída na lista crítica que pode ser visualizada no Tivoli Dynamic Workload Console.

Tabela 14. Opções Globais de Garantia de Serviço de Carga de Trabalho (continuação)

Opção	Descrição
approachingLateOffset | al	O horário de início crítico de uma tarefa na rede crítica é o horário mais recente em que a tarefa pode ser iniciada sem fazer com que a tarefa crítica termine após o prazo final. Na maioria dos casos, uma tarefa iniciará bem antes do horário de início crítico, de forma que se sua execução levar mais tempo que a duração estimada, a situação não se tornará imediatamente crítica. Portanto, se uma tarefa não for iniciada e o horário de início crítico estiver apenas alguns minutos atrasado, a conclusão adequada da tarefa crítica será considerada possivelmente um risco. A opção `approachingLateOffset` permite determinar o período de tempo antes do horário de início crítico de uma tarefa na rede crítica em que você é alertado para esse possível risco. Se uma tarefa ainda não tiver iniciado o número de segundos especificado antes do horário de início crítico, a tarefa será incluída na lista de favoritos que pode ser visualizada no Dynamic Workload Console. O padrão é 120 segundos. **Nota:** O valor desse parâmetro é verificado regularmente. Não é necessário executar a JnextPlan para que as alterações sejam efetivadas.
deadlineOffset | do	Em geral, um prazo final deve ser especificado para uma tarefa sinalizada como `critical`. Se não for, o planejador utilizará o prazo final definido para o fluxo de tarefas. A opção `deadlineOffset` fornece um deslocamento utilizado para calcular o horário de início crítico no caso do prazo final não existir para uma tarefa crítica e seu fluxo de tarefas. O fim do plano mais esse deslocamento são assumidos como o prazo final da tarefa crítica. O deslocamento é expresso em minutos. O padrão é 2 minutos. **Importante:** Quando o plano é estendido, o horário de início de tarefas críticas com um prazo final calculado com esse mecanismo é alterado automaticamente como conseqüência do fato que ele deve agora corresponder ao horário de encerramento do novo plano.

Para obter informações adicionais sobre opções globais, consulte *Guia de Administração do IBM Tivoli Workload Scheduler*.

Opções locais

A garantia do serviço de carga de trabalho utiliza opções locais para controlar a alocação de prioridade de recursos do sistema para tarefas na rede crítica que devem ser promovidas para manter o prazo final crítico. A Tabela 15 mostra as opções locais utilizadas pelo recurso de garantia de serviço de carga de trabalho. Para configurar opções locais, edite o arquivo *twshome*`\localopts` em cada estação de trabalho em que as tarefas críticas estarão em execução. Execute JnextPlan ou reinicie o agente para que as alterações feitas nas opções locais tomem efeito.

Tabela 15. Opções Locais de Garantia de Serviço de Carga de Trabalho

Opção	Descrição
jm promoted nice	Configura o valor `nice` para ser designado a tarefas críticas ou predecessores de tarefas críticas que precisam ser promovidos nos sistemas operacionais UNIX e Linux, para que sejam designados mais recursos para eles e eles sejam processados antes das outras tarefas. Valores específicos variam para as diferentes plataformas, mas, no geral, a configuração deve ser um número inteiro negativo. O padrão é -1 e números mais baixos representam prioridades mais altas. Se você especificar um número inteiro positivo, o valor padrão será utilizado. A opção local **jm nice** tem uma função similar na priorização de tarefas que foram enviadas pelo usuário root. Uma tarefa crítica que foi enviada pelo usuário root poderá ser elegível para os dois mecanismos de priorização. Nesse caso, os valores serão incluídos juntos. Por exemplo, se **jm promoted nice** for configurado para -4 e **jm nice** for -2, a tarefa crítica enviada pelo usuário root terá uma prioridade -6.

Tabela 15. Opções Locais de Garantia de Serviço de Carga de Trabalho (continuação)

Opção	Descrição
`jm promoted priority`	Configura o valor de prioridade para tarefas críticas ou predecessores de tarefas críticas que precisam ser promovidos para que sistemas operacionais Windows designem mais recursos para ele e os processem antes das outras tarefas. Os valores possíveis são: • Alto • AboveNormal • Normal • BelowNormal • Low ou Idle O padrão é AboveNormal. Observe que, se você configurar um valor de prioridade menor que um que pode ser designado para tarefas não críticas, nenhum aviso será fornecido e nenhum mecanismo como aquele disponível para jm promoted nice deverá ser configurado novamente como o padrão.

Requisitos do Arquivo de Segurança

É obrigatório que os usuários que possuem as instâncias do Tivoli Workload Scheduler que executam tarefas críticas sejam autorizados para trabalhar com todas as tarefas, fluxos de tarefa e estações de trabalho associados a estas tarefas. Estes usuários devem, portanto, ter os direitos DISPLAY, MODIFY e LIST no arquivo de segurança para todos os objeto associados a JOB, SCHEDULE e CPU.

Consulte o *Guia de Administração* do Tivoli Workload Scheduler para obter detalhes sobre o arquivo de segurança.

Planejando Tarefas Críticas

A garantia de serviço de carga de trabalho fornece um meio de identificar tarefas críticas, definir prazos finais e calcular horas para todas as tarefas que devem preceder a tarefa crítica.

Se for imprescindível que uma tarefa seja concluída antes de um horário específico, você poderá sinalizá-la como crítica ao incluí-la em um fluxo de tarefas utilizando as funções do Workload Designer no Dynamic Workload Console. Você pode definir o prazo final no nível da tarefa ou do fluxo de tarefas.

As tarefas também podem ser sinalizadas como críticas pela inclusão da palavra-chave **critical** na instrução de tarefa quando um fluxo de tarefas é criado ou modificado usando a linha de comandos do **composer**.

Quando o comando **JnextPlan** é executado para incluir a nova tarefa no plano de produção, todas as tarefas que são predecessoras diretas ou indiretas da tarefa crítica são identificadas. Estas tarefas, juntamente com a tarefa crítica em si, formam uma `rede crítica`.

Como a sincronização de tarefas na rede crítica deve ser rigidamente controlada, o Planejador de Tempo calcula as seguintes avaliações de desempenho para cada tarefa da rede crítica:

Início Crítico
Ele aplica-se apenas a sistemas distribuídos e representa a hora mais recente em que a tarefa pode ser iniciada sem fazer com que a tarefa crítica perca seu prazo final.

Os horários de início críticos são calculados iniciando com o prazo final configurado para a tarefa crítica e trabalhando para trás utilizando a duração estimada de cada tarefa para determinar seu horário de início crítico. Por exemplo, se o prazo final da tarefa crítica for 19h e a duração estimada da tarefa crítica for 30 minutos, a tarefa crítica não será concluída dentro do prazo final, a menos que tenha começado às 18h. Se o predecessor imediato da tarefa crítica tiver uma duração estimada de 20 minutos, ele deverá começar no máximo às 18h10.

Nota: Apenas o prazo final da tarefa crítica é considerado quando se calcula horários de início críticos para as tarefas na rede crítica. Se outras tarefas tiverem prazos finais definidos, suas horas de início críticas deverão ser posteriores aos seus prazos finais.

Início Antecipado
Representa a hora mais antiga em que uma tarefa na rede crítica poderia começar, levando em consideração todos os requisitos de dependências e de recursos.

Horários de início e de encerramento estimados
As horas de início estimadas são calculadas começando com a hora mais antiga em que a primeira tarefa ou tarefas na rede crítica poderiam começar, e trabalhando na dianteira utilizando a duração estimada de cada tarefa para estimar a hora de início da tarefa que segue.

Horários de início e de encerramento planejados
Para os cálculos iniciais, estes valores são configurados nos horários de início e de encerramento estimados. Eles são recalculados subseqüentemente para levar em consideração cada alteração ou atraso no plano.

Duração prevista
A duração estimada de uma tarefa é baseada nas estatísticas coletadas de execuções anteriores da tarefa. Se a tarefa nunca foi executada antes, é utilizado um valor padrão de um minuto. Leve-o em conta ao considerar a exatidão das horas calculadas para as redes de tarefas críticas que incluem tarefas sendo executadas pela primeira vez. NO caso de uma tarefa sombra, a duração estimada é sempre configurada com o valor padrão de um minuto. Isso se aplica às tarefas sombra sendo executadas pela primeira vez, assim como qualquer execução subsequente da tarefa sombra.

As horas de cada tarefa na rede crítica são incluídas no arquivo `Symphony` que inclui todas as informações do plano e que é distribuído para todas as estações de trabalho onde as tarefas devem ser executadas.

Conforme o plano é executado, o Monitor do Plano monitora todas as redes críticas: alterações subseqüentes na rede crítica que afetam a sincronização das tarefas acionarão o recálculo dos horários de início críticos e estimados. As mudanças podem incluir mudanças manuais, por exemplo, liberar dependências ou reexecutar tarefas, e mudanças feitas automaticamente pelo sistema em resposta a um risco potencial ou real para a conclusão em tempo hábil da tarefa crítica.

Visualizações específicas para tarefas críticas e seus predecessores, disponíveis no Dynamic Workload Console, permitem que você monitore o processamento da rede crítica. As visualizações podem identificar problemas imediatamente no planejamento da tarefa crítica. Por exemplo, se a hora de início estimada de uma

tarefa na rede crítica for posterior à hora de início crítica, isso será imediatamente sinalizado como um possível risco para a tarefa crítica.

Processando e Monitorando Tarefas Críticas

A garantia de serviço de carga de trabalho fornece rastreamento automático e priorização de tarefas de rede críticas e funções online que são utilizadas para monitorar e intervir no processamento de tarefas de rede críticas.

Rastreamento e Priorização Automáticas

Para garantir que prazos finais críticos possam ser cumpridos, a garantia de serviço de carga de trabalho fornece os seguintes serviços automatizados para tarefas críticas e tarefas predecessoras que formam suas redes críticas:

Promoção

Quando a hora de início crítica de uma tarefa está se aproximando e a tarefa ainda não começou, o mecanismo de promoção é utilizado. Uma tarefa promovida recebe recursos de sistema operacional adicionais e seu envio tem prioridade.

O tempo das promoções é controlado pela opção global `promotionoffset`. Tarefas promovidas são selecionadas para envio após as tarefas que têm prioridades "alta" e "ir", mas antes de todas as outras tarefas. A priorização de recursos do sistema operacional é controlada pelas opções locais `jm promoted nice` (UNIX e Linux) e `jm promoted priority` (Windows).

Cálculo do caminho crítico

O caminho crítico é a cadeia de dependências, levando à tarefa crítica, que tem o maior risco de perder o prazo final em qualquer momento fornecido. O caminho crítico é calculado utilizando horários de encerramento estimados dos predecessores de tarefa crítica. Voltando a partir da tarefa crítica, o caminho é construído selecionando o predecessor com o horário de encerramento estimado mais recente. Se o horário de encerramento real for substancialmente diferente do horário de encerramento estimado, o caminho crítico será automaticamente recalculado.

Figura 21 na página 109 mostra o caminho crítico através de uma rede crítica em um momento específico durante o processamento do plano.

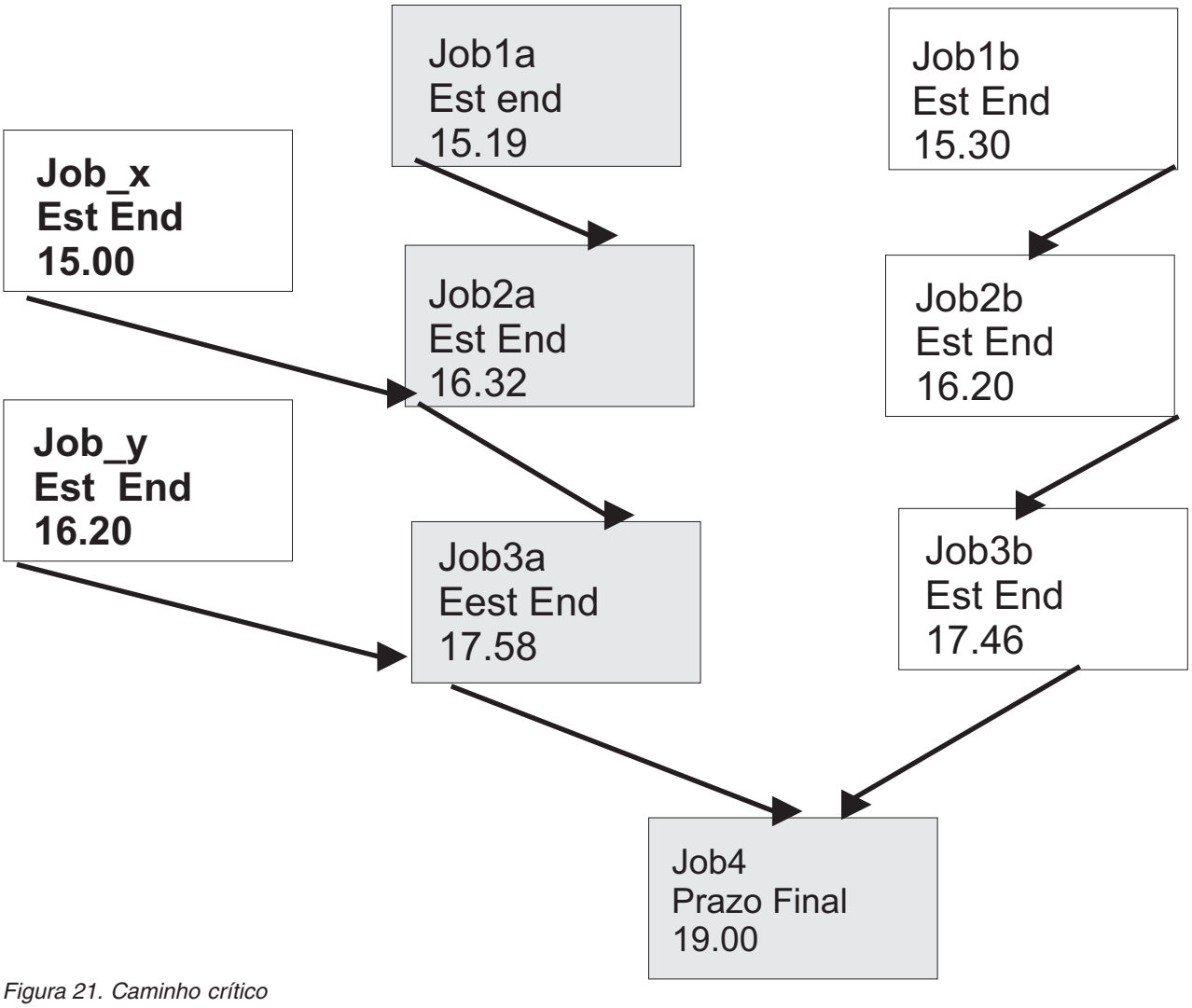


Figura 21. Caminho crítico

Neste momento, o caminho crítico inclui Job3a, Job2a e Job1a. Job3a e Job3b são os predecessores imediatos da tarefa crítica, Job4 e Job3a têm a data de encerramento estimada mais recente. Job3a tem dois predecessores imediatos, Job2a e Job_y. Job2a tem o horário de encerramento mais recente, e assim por diante.

Inclusão de tarefas na lista crítica

Tarefas que fazem parte da rede crítica são incluídas em uma lista crítica que é associada à tarefa crítica em si. A lista crítica inclui quaisquer tarefas de rede críticas que têm um impacto real ou potencial na conclusão em tempo hábil da tarefa crítica. Tarefas são incluídas na lista crítica por uma ou mais das razões listadas a seguir. Observe que apenas as tarefas que começam a rede crítica atual, para a qual não há um predecessor, podem ser incluídas na lista crítica.

- A tarefa parou com um erro. A duração do tempo antes do horário de início crítico é determinada pela opção global `approachingLateOffset`.
- A tarefa ficou em execução mais tempo do que o estimado por um fator definido na opção global `longDurationThreshold`.
- A tarefa ainda não foi iniciada, embora todas as suas dependências de seqüências tenham sido resolvidas ou liberadas, e pelo menos uma das condições a seguir é verdadeira:

- A hora de início crítica foi quase atingida.
- A tarefa é planejada para execução em uma estação de trabalho onde o limite é configurado como zero.
- A tarefa pertence a um fluxo de tarefas para o qual o limite está configurado como zero.
- A tarefa ou seu fluxo de tarefas foram suprimidos.
- A tarefa ou seu fluxo de tarefas têm atualmente uma prioridade inferior à fence ou estão configurados como zero.

Configuração de status de risco alto ou potencial para a tarefa crítica
Um status de risco pode ser configurado para a tarefa crítica da seguinte forma:

Risco Alto
Os tempos calculados mostram que a tarefa crítica será concluída antes de seu prazo final.

Risco Potencial
Tarefas predecessoras críticas foram incluídas na lista crítica.

Rastreamento Online de Tarefas Críticas

O Dynamic Workload Console fornece visualizações especializadas para rastrear o progresso de tarefas críticas e seus predecessores. É possível acessar as visualizações a partir do Painel ou criando tarefas Procurar Tarefas Críticas.

A visualização inicial lista todas as tarefas críticas para o mecanismo, mostrando o status: normal, risco potencial ou risco alto. Nesta visualização, é possível navegar para ver:

- A lista especial de tarefas que colocam o prazo final crítico em risco.
- O caminho crítico.
- Detalhes de todos os predecessores críticos.
- Detalhes dos predecessores críticos concluídos.
- Logs de tarefas que já foram executadas.

Utilizando as visualizações, você pode monitorar o progresso da rede crítica, descobrir problemas atuais e potenciais, liberar dependências e reexecutar tarefas.

Cenário de Garantia de Serviço de Carga de Trabalho

Este cenário ilustra o uso da garantia de serviço de carga de trabalho para garantir que prazos finais de produção importantes possam ser cumpridos.

A Fine Cola utiliza o Tivoli Workload Scheduler para gerenciar a sincronização e as interdependências de seu processo de produção e fornecimento.

A Fine Cola possui acordos de nível de serviço com vários clientes que suportam a reestocagem "sob demanda". Isto significa que inícios postergados em qualquer uma das rotas de entrega certamente resultarão na falta de produtos da Fine Cola nas prateleiras.

A tarefa que produz os pedidos de carregamento para caminhões deve ser concluída no máximo às 5h30. Esta tarefa depende da conclusão bem-sucedida de outras tarefas. Por exemplo, embora pedidos sejam processados antes da hora, mudanças de última hora acontecem quando os caminhões retornam após concluir

uma rota de entrega. A Fine Cola também fornece faturas com notas de entrega, portanto as mudanças nos pedidos também devem ser refletidas nos preços e podem acionar ajustes de oferta especial nos preços.

Planejando a Tarefa Crítica

Usando o Workload Designer no Dynamic Workload Console, o planejador Fine Cola sinaliza a tarefa de ordem de carregamento como crítica e configura o prazo final para 5 horas das manhã.

Quando**JnextPlan** é executado, as datas de início críticas para esta tarefa e todas as tarefas identificadas como predecessoras da tarefa crítica são calculadas.

Rastreando a Tarefa Crítica

1. O operador do Tivoli Workload Scheduler verifica os painéis e vê que existem tarefas críticas planeadas em um dos mecanismos.
2. Ele vê que há uma tarefa crítica em risco potencial. Ele clica no link de risco potencial para obter a lista de tarefas críticas nesse estado.

   A tarefa de pedidos de carregamento mostra um status de "possível risco".
3. Ele seleciona a tarefa e clica em **Lista de Favoritos** para a ver a tarefa ou tarefas que estão colocando a tarefa crítica em risco.

   A tarefa de ajuste de pedidos é listada como estando com erro.
4. Ele seleciona a tarefa e clica em **Log de Tarefa**.

   O log mostra que a tarefa falhou devido a credenciais incorretas para o banco de dados de pedidos.
5. Após descobrir que a senha do banco de dados foi alterada nesse dia, ele altera a definição de tarefa no arquivo symphony e reexecuta a tarefa.
6. Quando ele retorna ao painel, vê que não há mais nenhuma tarefa em risco potencial. Além disso, a lista de tarefas críticas que foi aberta ao clicar no link de risco potencial não mostra mais a tarefa crítica após a tarefa ser reexecutada.
7. A tarefa está agora em execução e foi promovida automaticamente de forma a receber uma prioridade mais alta para envio e recursos do sistema.
8. Nenhum problema adicional precisa de correção e a tarefa crítica finalmente é concluída às 4h45.

Capítulo 6. Customizando a Carga de Trabalho Utilizando Tabelas de Variáveis

Este capítulo introduz o conceito de tabelas de variáveis para agrupar parâmetros globais, daqui em diante chamadas de variáveis, para customizar sua carga de trabalho.

A partir do Tivoli Workload Scheduler versão 8.5, o que era chamado de parâmetros globais nas versões anteriores, agora são chamados de variáveis. As definições das variáveis estão contidas nas tabelas de variáveis. Uma tabela de variáveis é um objeto que agrupa várias variáveis. Utilizando tabelas de variáveis, é possível designar diferentes valores para a mesma variável para utilizar nas definições de tarefa e de fluxo de tarefas na JCL, logon, dependências de avisos, dependências de arquivos e prompts de recuperação. Isto é particularmente útil quando uma definição de tarefa é utilizada como modelo para uma tarefa que pertence a mais de um fluxo de tarefa. Por exemplo, é possíveld designar valores diferentes para a mesma variável e reutilizar a mesma definição de tarefa em fluxos de tarefas diferentes, economizando tempo e dinheiro.

Quando você define uma variável, você a designa a uma tabela de variáveis porque a mesma variável pode ser definida em diferentes tabelas de variáveis com diferentes valores. Ou uma melhor abordagem é criar uma ou mais tabelas de variáveis, especificando uma lista de nomes de variáveis e valores para cada tabela. Ao fazer isto, é possível incluir o mesmo nome de variável com diferentes valores em diferentes tabelas. Ao solicitar uma lista de variáveis, você obtém pares *variabletable.variablename* para identificar facilmente a que tabela de variáveis a variável pertence.

Por exemplo, a variável VAR1 definida na tabela de variáveis REP1_TABLE1 é mostrada como:

```
REP1_TABLE1.VAR1
```

É possível designar tabelas de variáveis para ciclos de execução, fluxos de tarefas e estações de trabalho.

Com tarefas de variáveis, você altera o comportamento da carga de trabalho de acordo com quando, porque e onde deseja executar seu planejamento, dando a você uma maior flexibilidade para customizar sua carga de trabalho e para cumprir os acordos de nível de serviço. Em detalhe:

Quando
: Para alterar o comportamento de tarefas e fluxos de tarefas com base em quando elas estão planejadas para executar, isto é, em quais dias elas são executadas. Utilizando tabelas de variáveis com ciclos de execução.

Finalidade
: Para alterar o comportamento de tarefas e fluxos de tarefas com base na finalidade de elas serem planejadas para executar, por exemplo, para criar uma tarefa que executa comandos diferentes. Utilizando tabelas de variáveis com fluxos de tarefas.

Onde
: Para alterar o comportamento de tarefas e fluxos de tarefas com base no local onde elas são executadas, por exemplo, em estações de trabalho diferentes. Utilizando tabelas de variáveis com estações de trabalho.

© Copyright IBM Corp. 1999, 2014

Migrando Parâmetros Globais de Versões Anteriores

Considerações ao Migrar Parâmetros Globais de Versões Anteriores

Ao fazer upgrade de versões anteriores à 8.5, as definições de parâmetro global, agora chamadas de definições de variáveis, que você possui no banco de dados, são automaticamente migradas para a tabela de variáveis padrão chamada MAIN_TABLE. Após a atualização:

- Todas as variáveis são precedidas pelo nome de tabela padrão. Por exemplo, após a migração, a variável *REP_PATH* receberá o seguinte nome:

  ```
  MAIN_TABLE.REP_PATH
  ```

  Ao solicitar uma lista de variáveis, você obtém pares *variabletable.variablename* para identificar facilmente a que tabela de variáveis a variável pertence.
- Sua carga de trabalho é resolvida da mesma forma que antes da migração, porque qualquer objeto do Tivoli Workload SchedulerTivoli Workload Scheduler contendo variáveis refere-se a MAIN_TABLE para a resolução de variável.
- Para cada seção do usuário que inclui a palavra-chave `parameter`, a seguinte linha é incluída no arquivo de segurança:

  ```
  vartable name=@ access=add,delete,display,modify,list,use,unlock
  ```

Para obter detalhes sobre o processo de upgrade, consulte Tivoli Workload Scheduler*Tivoli Workload Scheduler*.

Ao fazer upgrade da versão 8.3 ou mais recente, não modifique variáveis até que tenha migrado o gerenciador de domínio principal e todos os seus principais de backup porque, nesta fase de transição, você tem duas versões diferentes do banco de dados. Se você tiver que incluir ou modificar variáveis durante esta fase de transição, certifique-se de fazer a mudança na versão 8.3 ou 8.4 e na versão 8.5 dos gerenciadores de domínio principal.

Os parâmetros locais que foram criados e gerenciados com o comando utilitário **parms** no banco de dados de parâmetro local nas estações de trabalho funcionam da mesma maneira que antes.

Na V8.3 e V8.4, os parâmetros são armazenados na tabela MDL.VAR_VARIABLES do banco de dados. Após o upgrade para a V8.5 ou mais recente, os mesmos parâmetros são armazenados no MAIN_TABLE contido na tabela de banco de dados MDL.VAR_VARIABLES2. Se você começou a migrar seu ambiente, mas ainda não concluiu a migração, e o gerenciador de domínio principal é V8.3 ou V8.4 e o gerenciador do domínio backup principal é V8.5 ou mais recente, ou vice-versa, o gerenciador de domínio principal V8.3 ou V8.4 não reconhecerá a tabela MDL.VAR_VARIABLES2, portanto, se você alterar um parâmetro no gerenciador de domínio principal V8.3 ou V8.4, esta mudança será armazenada na tabela antiga (MDL.VAR_VARIABLES). Se você alterar o parâmetro no gerenciador de domínio principal V8.5, a mudança será armazenada na tabela MDL.VAR_VARIABLES2.

A Tabela de Variáveis Padrão

Este tópico descreve a tabela de variáveis padrão e como ela funciona.

A tabela de variáveis padrão é a tabela que contém todas as variáveis que você definiu sem especificar nenhum nome de tabela de variáveis. O nome padrão da tabela de variáveis padrão é **MAIN_TABLE**. É possível modificar esse nome a

qualquer momento ou configurar outra tabela de variáveis como a tabela de variáveis padrão. Não é possível excluir a tabela de variáveis padrão. Quando você configura outra tabela de variáveis como o padrão, a tabela de variáveis padrão original não é mais marcada como padrão. É possível trabalhar com a tabela de variáveis padrão da mesma maneira que com qualquer outra tabela de variáveis. É possível identificar facilmente a tabela de variáveis padrão na interface com o usuário, pois ela é marcada com um **Y** no campo **default**.

Exemplo

Este exemplo mostra uma lista de tabelas de variáveis

```
Variable Table Name      Default Updated On Locked By
---------------------------------------  -------  ----------  ----------------
MAIN_TABLE         Y  05/07/2008 -
VT_1              05/07/2008 -
VT_2              05/07/2008 -
VARTABLE3            05/07/2008 -
V4               05/07/2008 -
VT5               05/07/2008 -

AWSBIA291I Total objects: 6
```

Integridade de Dados para Tabelas de Variáveis

Este tópico explica como a integridade de dados é garantida utilizando as tabelas de variáveis.

Assim como ocorre com os outros objetos, o Tivoli Workload Scheduler mantém a integridade de dados da tabela de variáveis sempre que você executa comandos que criam, modificam, renomeiam ou excluem a definição de uma tabela de variáveis.

Ao referenciar uma tabela de variáveis a partir de qualquer ciclo de execução, fluxo de tarefas ou estação de trabalho, o Tivoli Workload Scheduler verifica se a tabela de variáveis existe e preserva o link entre ela e o ciclo de execução, fluxo de tarefas e estação de trabalho. Isto significa que você não pode excluir uma definição de tabela de variáveis enquanto uma referência de um ciclo de execução, um fluxo de tarefa ou uma estação de trabalho ainda existe.

A integridade das referências é garantida no nível da tabela de variáveis e não no nível de variável, isto é, quando um ciclo de execução, um fluxo de tarefas e uma estação de trabalho faz referência a uma tabela de variáveis, o Tivoli Workload Scheduler verifica se essa tabela existe, não se a variável mencionada existe.

Mecanismo de Bloqueio para Tabelas de Variáveis

Este tópico descreve como o mecanismo de bloqueio funciona para tabelas de variáveis.

O bloqueio é configurado no nível de tabela de variáveis para assegurar que as definições no banco de dados não sejam sobrescritas por diferentes usuários que estão acessando simultaneamente a mesma tabela de variáveis. Isto significa que você obtém permissões exclusivas para todas as variáveis nessa tabela tanto quando bloqueia uma tabela de variáveis como quando bloqueia uma variável. Isto é, é possível executar quaisquer comandos na tabela de variáveis bloqueada e em todas as variáveis contidas nela. Qualquer outro usuário tem apenas acesso de leitura a essa tabela de variáveis e às variáveis contidas nela.

Isto impede que qualquer outro usuário altere a mesma tabela de variáveis que você está alterando. Se outro usuário tentar bloquear uma tabela de variáveis ou uma variável que você já tenha bloqueado, uma mensagem de erro será retornada.

Segurança da Tabela de Variável

Este tópico descreve como definir configurações de segurança para tabelas de variáveis.

O acesso do usuário a tabelas de variáveis deve ser autorizado no arquivo de segurança do Tivoli Workload Scheduler. Assim como para outros objetos, o conector verifica a existência de autorização apropriada antes de executar uma ação que requeira acesso a uma tabela de variáveis. A nova palavra-chave a seguir está disponível no arquivo de segurança para esta finalidade:

```
vartable name=@ access=add,delete,display,modify,list,use,unlock
```

É necessário o acesso `use` para poder referenciar uma tabela de variáveis de outros objetos (fluxos de tarefa, ciclos de execução e estações de trabalho). Os filtros de segurança são baseados apenas no atributo`name`, mas seu administrador do Tivoli Workload Scheduler tem a opção de utilizar a palavra-chave **$default** para especificar permissões de segurança na tabela padrão, independentemente de seu nome.

A permissão para trabalhar em uma variável não é mais baseada na variável individual mas na tabela que a contém. O acesso a uma variável é concedido apenas se a ação correspondente na tabela de variáveis que a inclui for permitida. A tabela a seguir mostra as permissões correspondentes para variáveis e tabelas de variáveis:

Tabela 16. O Relacionamento entre as Tabelas de Variáveis em suas Variáveis no Arquivo de Segurança do Tivoli Workload Scheduler

Acesso definido à tabela de variáveis	Ação permitida nas variáveis contidas
modify	add
	excluir
	modify
display	display
destravar	destravar

A partir da versão 8.5, a palavra-chave `parameter` no arquivo de segurança se aplica apenas a parâmetros locais.

Consulte o Guia de Administração do *Tivoli Workload Scheduler* para obter detalhes sobre o arquivo de segurança.

Resolução de Variável

Este tópico descreve como as variáveis são resolvidas ao gerar um plano e ao enviar uma tarefa ou um fluxo de tarefas para ser executado.

O formato usado para especificar uma variável também determina quando uma variável é resolvida com um valor. Consulte “Definição de Variável e Parâmetro” na página 198 para obter informações sobre os formatos possíveis de usar.

Quando você gera um plano, o Tivoli Workload Scheduler analisa as tabelas de variáveis na ordem mostrada a seguir para resolução da variável:

1. No ciclo de execução. O ciclo de execução no fluxo de tarefas é verificado primeiro, então o ciclo de execução em um grupo de ciclo de execução e finalmente, a tabela de variável definida no nível do grupo de ciclo de execução.
2. No fluxo de tarefas.
3. Na estação de trabalho. Consulte "Estação de Trabalho Considerada para Resolução de Variáveis".
4. Na tabela de variável padrão, mas apenas quando as variáveis são especificadas no formato ^*variablename*^ na definição de tarefa. As variáveis especificadas no formato ${*variablename*} não são resolvidas.

No momento de resolução do plano, cada nível é analisado na ordem descrita acima. Se você especificar uma variável que não estiver contida em nenhuma tabela de variável, incluindo a padrão, uma mensagem de aviso contendo o nome da variável não resolvida será gravada no arquivo de log *<WAS_profile_path>*\logs\twaserver\SystemOut.log e o nome da variável fica no plano, em que o caminho padrão para o *<WAS_profile_path>* é *<TWA_home>*/WAS/TWSprofile.

Quando você envia um fluxo de tarefas, o Tivoli Workload Scheduler determina variáveis analisando as tabelas de variáveis na ordem mostrada a seguir:
1. Especificada durante a operação de envio.
2. No fluxo de tarefas.
3. Na estação de trabalho. Consulte "Estação de Trabalho Considerada para Resolução de Variáveis".
4. Na tabela de variável padrão, mas apenas quando as variáveis são especificadas no formato ^*variablename*^ na definição de tarefa. As variáveis especificadas no formato ${*variablename*} não são resolvidas.

Quando você envia uma tarefa, o Tivoli Workload Scheduler determina variáveis analisando as tabelas de variáveis na ordem mostrada a seguir:
1. Especificada durante a operação de envio, mas apenas quando as variáveis são especificadas no formato ^*variablename*^ na definição de tarefa. As variáveis especificadas no formato ${*variablename*} não são resolvidas.
2. Na estação de trabalho. Consulte "Estação de Trabalho Considerada para Resolução de Variáveis".
3. Na tabela de variável padrão, mas apenas quando as variáveis são especificadas no formato ^*variablename*^ na definição de tarefa. As variáveis especificadas no formato ${*variablename*} não são resolvidas.

Estação de Trabalho Considerada para Resolução de Variáveis

Quando a variável é determinada pela tabela de variáveis especificada na estação de trabalho, a estação de trabalho levada em consideração é:

Para variável na dependência de arquivo
A estação de trabalho em que o arquivo reside.

Para variável na tarefa
A estação de trabalho onde a tarefa é definida.

Para variável na dependência de aviso

Prompt Global
Nenhuma estação de trabalho é levada em consideração. As variáveis em prompts globais são determinadas sempre utilizando a tabela de variáveis padrão. Isso ocorre porque o prompt global é utilizado por todas as tarefas e fluxos de tarefas, portanto, apenas um valor deve ser utilizado para a resolução da variável.

Prompt ad hoc
A estação de trabalho na qual a tarefa ou o fluxo de tarefas que depende da dependência de aviso é definida.

Capítulo 7. Automação de Carga de Trabalho Orientada pelo Evento em Execução

A automação de carga de trabalho orientada por evento inclui a capacidade de executar a automação de carga de trabalho On Demand, além do planejamento de tarefa baseado em plano. Fornece a capacidade de definir regras que podem acionar a automação de carga de trabalho On Demand.

O objeto da automação de carga de trabalho orientada por evento no Tivoli Workload Scheduler é executar um conjunto predefinido de ações em resposta a eventos que ocorrem nos nós em que o Tivoli Workload Scheduler é executado (mas também em nós não Tivoli Workload Scheduler, quando a linha de comandos `sendevt` é utilizada). Isso implica na capacidade de enviar a carga de trabalho e executar comandos on the fly, notificar usuários por meio de e-mail ou enviar mensagens ao Tivoli Enterprise Console.

As principais tarefas da automação de carga de trabalho orientada por evento são:

- Acionar a execução de tarefas em lote e fluxos de tarefas com base na recepção ou combinação de eventos em tempo real.
- Responder a avisos
- Notificar os usuários quando condições anormais ocorrem no ambiente de planejamento do Tivoli Workload Scheduler ou na atividade de planejamento em lote.
- Chamar um produto externo quando uma condição de evento específica ocorre.

A automação de carga de trabalho orientada por evento é baseada no conceito de regra de evento. No Tivoli Workload Scheduler, uma regra de evento é um objeto de planejamento que inclui os seguintes itens:

- Eventos
- Condições de correlação de eventos
- Ações

Quando você definir uma regra de evento, especifique um ou mais eventos, uma regra de correlação e uma ou mais ações que sejam acionadas por esses eventos. Além disso, é possível especificar datas de validades, um intervalo de tempo diário da atividade e um fuso horário comum para todas as restrições de tempo configuradas.

Os eventos que o Tivoli Workload Scheduler pode detectar para o acionamento da ação podem ser:

Eventos Internos
: Eles são eventos que envolvem o status do aplicativo interno do Tivoli Workload Scheduler e alterações no status dos objetos do Tivoli Workload Scheduler. Eventos desta categoria podem ser alterações de status de tarefas ou fluxos de tarefas, tarefas ou fluxos de tarefas críticos sofrendo atrasos ou cancelamentos e alterações de status de estações de trabalho.

Eventos Externos
: Eles são eventos que não envolvem diretamente o Tivoli Workload Scheduler que podem impactar, entretanto, o envio de carga de trabalho.

© Copyright IBM Corp. 1999, 2014

Eventos desta categoria podem ser mensagens gravadas em arquivos de log, eventos enviados por aplicativos de terceiros ou um arquivo sendo criado, atualizado ou excluído.

Em uma regra, dois ou mais eventos podem ser correlacionados por meio de atributos de correlação, tais como uma estação de trabalho ou tarefa comum. Os atributos de correlação fornecem uma maneira de direcionar a regra para criar uma regra separada (ou cópia de si mesma) para cada grupo de eventos que compartilham características comuns. Em geral, cada regra ativa possui uma cópia em execução no servidor de processamento de eventos. Entretanto, às vezes, a mesma regra é necessária para diferentes grupos de eventos, que geralmente estão relacionados a diferentes grupos de recursos. Utilizar um ou mais atributos de correlação é um método de direcionar uma regra para criar uma cópia da regra separada para cada grupo de eventos com características comuns.

As ações que o Tivoli Workload Scheduler pode executar quando detecta qualquer um destes eventos podem ser:

Ações Operacionais
Elas são ações que causam a alteração no status dos objetos do planejamento. Ações desta categoria são enviar uma tarefa, fluxo de tarefas ou comando, ou responder a um aviso.

Ações de Notificação
Elas são ações que não possuem impacto no status dos objetos de planejamento. As ações pertencentes a esta categoria estão enviando um e-mail, registrando o evento em um banco de dados de auditoria interna, encaminhando o evento ao Tivoli Enterprise Console ou executando um comando que não seja do Tivoli Workload Scheduler.

Esta classificação de eventos e ações é conceitual. Ela não influencia na maneira que eles são tratados pelo mecanismo acionado por eventos.

Cenários Simples de Regras de Eventos

Esta seção lista alguns cenários simples que envolvem o uso de regras de eventos. A codificação XML correspondente é mostrada na "Exemplos de Regras de Eventos" na página 128.

Cenário 1: Envia notificação por e-mail

1. O administrador define a seguinte regra de evento:
   - Quando alguma das tarefas `job123` finaliza com erro e gera a seguinte mensagem de erro:

     ```
     AWSBHT001E The
     job "MYWORKSTATION#JOBS.JOB1234" in file "ls" has
     failed with the error: AWSBDW009E The following operating system
     error occurred retrieving the password structure for either the
     logon user...
     ```

     enviar um e-mail ao operador `john.smith@mycorp.com`. O assunto do e-mail inclui os nomes da instância da tarefa e da estação de trabalho associada.

     A regra de evento é válida de 1° de Dezembro a 31 de Dezembro no espaço de tempo 12h-16h, horário do leste.
2. O administrador salva a regra como não rascunho no banco de dados e ela é prontamente implementada pelo Tivoli Workload Scheduler.
3. O planejador inicia o monitoramento das tarefas e toda vez que uma delas termina com erro, John Smith recebe um e-mail para que ele possa verificar a tarefa e tome uma ação corretiva.

Cenário 2: Monitor que a estação de trabalho vincula de volta

1. O administrador define a seguinte regra de evento:
   - Se o link da estação de trabalho `CPU1` for removido e o link não for reestabelecido dentro do período de 10 minutos, envia uma notificação por e-mail para `chuck.derry@mycorp.com`.
2. O administrador salva a regra como não rascunho no banco de dados e ela é prontamente implementada pelo Tivoli Workload Scheduler.
3. O planejador inicia o monitoramento de `CPU1`.

   Se o status da estação de trabalho se tornar desvinculado, o Tivoli Workload Scheduler será iniciado dentro do tempo limite de 10 minutos. Se o evento `CPU1 linked` não for recebido dentro de 10 minutos, o planejador envia a notificação por e-mail a Chuck Derry.
4. Chuck Derry recebe o e-mail, consulta as ações/regras que foram acionadas nos últimos 10 minutos, e a partir daí navega até a instância `CPU1` e executa uma primeira análise de problema.

Cenário 3: Enviar o fluxo de tarefas quando o FTP tiver sido concluído

1. O administrador define a seguinte regra de evento:
   - Quando o arquivo `daytransac*` for criado no diretório `SFoperation` na estação de trabalho `system1` e as modificações no arquivo forem finalizadas, envie o fluxo de tarefas `calmonthlyrev`.

     A regra de evento é válida durante o ano no espaço de tempo das 18h às 22h no horário do leste.
2. O administrador salva a regra como não rascunho no banco de dados e ela é prontamente implementada pelo Tivoli Workload Scheduler.
3. O planejador inicia o monitoramento do diretório `SFoperation`. Assim que o arquivo `daytransac*` é criado e não está mais em utilização, ele envia o fluxo de tarefas `calmonthlyrev`.
4. O operador pode verificar os logs para localizar se a regra de evento ou o fluxo de tarefas foi executado.

Cenário 4:Iniciar tarefas de longa duração com base no tempo limite

1. O administrador define a seguinte regra de evento:
   - Quando o evento `job-x=exec` e o evento `job-x=succ/abend` são recebidos em 5 minutos, o planejador deverá responder com `Sim` ao `prompt-1` e iniciar o fluxo de tarefas `jobstream-z`, caso contrário ele deverá enviar um e-mail para `twsoper@mycompany.com` alertando que a tarefa está atrasada.
2. O administrador salva a regra de evento no status de rascunho. Após alguns dias, ele edita a regra, muda o destinatário de e-mail e a salva como não rascunho. A regra é implementada.
3. Toda vez que o status de `job-x` se torna `exec`, o Tivoli Workload Scheduler inicia o tempo limite de 5 minutos.

   Se o estado interno `job-x` não alterar para `succ` ou `abend` dentro de 5 minutos e o evento correspondente não for recebido, o Tivoli Workload Scheduler envia o e-mail, caso contrário ele reaplica `Sim` ao prompt e envia `jobstream-z`.

Cenário 5: Monitorando o status do processo e executando um script em lote

O administrador cria uma regra para monitorar o status de processos do Tivoli Workload Scheduler e executa um script em lote.

Cenário 6:Integração com o SAP R/3 (em combinação com o Tivoli Workload Scheduler for Applications)

1. O administrador define a seguinte regra de evento:
   - Quando um evento chamado `ID3965` é gerado no servidor SAP R/3 `Billing`, o Tivoli Workload Scheduler deve:
     a. Executar o comando:

```
"/usr/apps/helpDesk –openTicket –text
 'Processing error $parameter
  on SAP system $wsname'"
```

        para abrir um ticket do service desk
     b. Enviar um evento ao Tivoli Enterprise Console.
2. O administrador salva a regra como não rascunho e ela é implementada prontamente.
3. O Tivoli Workload Scheduler inicia o monitoramento do status de eventos do SAP R/3 ativados no sistema `Billing`.

   Quando o evento `ID3965` é detectado, o Tivoli Workload Scheduler executa o comando do help desk especificado e envia um evento ao TEC.
4. Após algum tempo, o administrador configura a regra de evento no status de rascunho. A regra é desativada automaticamente. Ela pode ser implementada novamente quando necessário.

Cenário 7: Monitorando o status do arquivo Symphony e registrando a ocorrência de um registro corrompido

O administrador cria uma regra para monitorar o status do arquivo Symphony na instância do Tivoli Workload Scheduler e registra a ocorrência de um registro de dependência Symphony corrompido no banco de dados de auditoria interna.

O Processo de Gerenciamento de Regras de Eventos

A automação da carga de trabalho orientada por evento é um processo contínuo e pode ser reduzida às seguintes etapas:

1. Uma definição de regra de evento é criada ou modificada com o Dynamic Workload Console ou com a linha de comandos do composer e salva no banco de dados de objetos. As definições de regra podem ser salvas como rascunho ou não rascunho.
2. Todas as regras não rascunho novas e modificadas salvas no banco de dados são periodicamente (por padrão, a cada cinco minutos) localizadas, construídas e implementadas por um processo interno denominado `construtor de regras`. Nesse momento, elas se tornam ativas. Entretanto, um `servidor de processamento de eventos`, que normalmente está localizado no gerenciador de domínio principal, recebe todos os eventos dos agentes e os processa.
3. As configurações de monitoramento atualizadas são transferidas por download para os agentes do Tivoli Workload Scheduler e ativadas. Cada agente do Tivoli Workload Scheduler executa um componente denominado `monman` que gerencia dois serviços denominados `mecanismo de monitoramento` e `ssmagent`, os quais devem capturar os eventos que ocorrem no agente e executar neles uma ação de filtragem preliminar.
4. Cada `monman` detecta e envia seus eventos para o servidor de processamento de eventos.
5. O servidor de processamento de eventos recebe os eventos e verifica se eles correspondem a qualquer regra de evento implementada.
6. Se uma regra de evento for correspondida, o servidor de processamento de eventos chamará um auxiliar de ações para executar as ações.

7. O auxiliar de ações cria uma instância de regra de evento e registra o resultado da ação no banco de dados.
8. O administrador ou o operador revisa o status das instâncias de regras de eventos e ações no banco de dados e logs.

O recurso de automação de carga de trabalho orientada por evento é automaticamente instalado com o produto. Você poderá alterar o valor da opção global `enEventDrivenWorkloadAutomation` a qualquer momento, caso não deseje utilizá-la na rede do Tivoli Workload Scheduler.

A automação da carga de trabalho orientada a evento é baseada em vários serviços, subsistemas e mecanismos internos. Os seguintes são significativos porque podem ser gerenciados:

monman

É instalado em cada agente do Tivoli Workload Scheduler no qual ele verifica todos os eventos locais. Todos os eventos detectados são redirecionados ao servidor de processamento de eventos. Os seguintes comandos `conman` estão disponíveis para gerenciar o `monman`:

Tabela 17. Comandos conman para Gerenciar Mecanismos de Monitoramento

Comando	Finalidade
deployconf	Atualiza o arquivo de configuração de monitoramento do mecanismo de monitoramento de eventos em um agente. É um comando opcional, uma vez que a configuração é normalmente implementada automaticamente.
showcpus getmon	Retorna a lista de regras de eventos definidas para o monitor em execução em um agente. Esse comando pode ser utilizado remotamente a fim de obter as informações do arquivo de configuração em outro agente da rede.
startmon	Inicia o `monman` em um agente. Pode ser emitido a partir de um agente diferente.
stopmon	Para o `monman` em um agente. Pode ser emitido a partir de um agente diferente.

`monman` é iniciado automaticamente toda vez que um novo Symphony é ativado. Isso é determinado pela opção local `autostart monman` que é configurada como `yes` por padrão (e que será possível desativar, se não desejar monitorar eventos em um determinado agente).

Após cada ciclo de implementação de regra, configurações de monitoramento atualizadas são automaticamente distribuídas aos agentes que hospedam as regras alteradas desde a última implementação. Observe que pode haver algumas situações transitórias enquanto a implementação está em curso. Por exemplo, se uma regra está pendente de desativação, os agentes podem estar enviando eventos na fração de tempo em que os arquivos de configuração novos ainda não estão implementados, mas o processador de eventos já os descarta.

Se um agente não puder enviar eventos ao `servidor de processamento de eventos` por um período de tempo especificado, o status de monitoramento do agente é automaticamente desligado. O período de tempo pode ser customizado (em segundos) com o parâmetro `edwa connection timeout` no arquivo `localopts`. Por padrão, ele é definido como 300 segundos (5 minutos).

Os eventos a seguir podem ser configurados no arquivo `BMEvents.conf` para postar o status de monitoramento de um agente:

- TWS_Stop_Monitoring (261) : enviado quando o status de monitoramento de um agente é configurado como off (para o comando stopmon ou porque o agente não pode enviar os eventos ao `servidor de processamento de eventos`).

- TWS_Start_Monitoring (262): enviado quando o status de monitoramento de um agente é configurado como on (para o comando startmon ou porque o agente reinicializou o envio de eventos ao `servidor de processamento de eventos`).

Esses eventos possuem os seguintes campos posicionais:

1. Número do Evento
2. Estação de trabalho afetada
3. Reservado, atualmente sempre configurado como 1

Servidor de processamento de eventos

Pode ser instalado no gerenciador de domínio principal, backup principal ou em qualquer agente tolerante a falhas instalado como backup principal. Ele é executado no servidor de aplicativos. Pode estar ativo em apenas um nó na rede. Ele constrói as regras, cria arquivos de configuração para os agentes e notifica os agentes para fazerem o download das novas configurações. Recebe e correlaciona os eventos enviados pelos mecanismos de monitoramento e executa as ações. Os seguintes comandos `conman` estão disponíveis para gerenciar o servidor de processamento de eventos:

Tabela 18. Comandos conman para Gerenciar o Servidor de Processamento de Eventos

Comando	Finalidade
starteventprocessor	Inicia o servidor de processamento de eventos
stopeventprocessor	Pára o servidor de processamento de eventos
switcheventprocessor	Comuta o servidor de processamento de eventos do gerenciador de domínio principal para o backup principal ou o agente tolerante a falhas instalado como um backup principal, ou vice-versa.

O servidor de processamento de eventos é iniciado automaticamente com o gerenciador de domínio principal. Somente um processador de eventos pode ser executado na rede a qualquer momento. Se você desejar executar o processador de eventos instalado em uma estação de trabalho que não seja a principal (isto é, no backup principal ou em qualquer agente tolerante a falhas instalado como o backup principal), primeiro utilize o comando switcheventprocessor para torná-lo o servidor de processamento de eventos ativo.

Nota: Se você configurar a palavra-chave **`ignore`** na definição de estação de trabalho do agente (instalado como backup principal) que no momento hospeda o processador de eventos ativo, a primeira ocorrência de **`JnextPlan`** a seguir reconhece que esse agente específico esteja fora do plano. Como conseqüência, não será possível reiniciar o processador de eventos hospedado nesse agente. Por esta razão, o planejador gera uma mensagem de aviso e inicia o processador de eventos hospedado pelo gerenciador de domínio principal.

Utilizando as Interfaces e Comandos Envolvidos

A execução e o gerenciamento da automação da carga de trabalho orientada a evento chamam as seguintes tarefas:

- Editar as definições de configuração
- Modelar regras de eventos
- Implementar ou desimplementar regras de eventos manualmente
- Gerenciar dispositivos de monitoramento e de processamento de eventos
- Monitorar e gerenciar instâncias de regras de eventos

Você deve estar pronto para utilizar diversas interfaces e comandos do Tivoli Workload Scheduler para realizá-las. A Tabela 19 resume os necessários:

Tabela 19. Interfaces e Comandos para Gerenciar a Automação de Carga de Trabalho Orientada a Evento

Interface ou Comando	Utilizar para...
optman	Alterar os valores padrão das opções globais associadas ao gerenciamento de eventos. As opções globais são utilizadas para configurar: • A freqüência com que as definições de regras são verificadas quanto à existência de atualizações (`deploymentFrequency`). As definições modificadas são implementadas no domínio do Tivoli Workload Scheduler • O número de porta EIF em que o servidor de processamento de eventos recebe eventos (`eventProcessorEIFPort` ou `eventProcessorEIFSSLPort` quando protegido com SSL). • O gerenciamento das políticas de limpeza de instância de regra, execução de ação e dados do log de mensagens (`logCleanupFrequency`). • As propriedades de servidor SMTP se forem implementadas regras com ações que enviam e-mails por meio de um servidor SMTP (`smtpServerName`, `smtpServerPort`, `smtpUseAuthentication`, `smtpUserName`, `smtpUserPassword`, `smtpUseSSL`, `smtpUseTLS`). • As propriedades do servidor Tivoli Enterprise Console, se você implementar regras que implementem ações que redirecionam eventos ao TEC (`TECServerName`, `TECServerPort`). • A possibilidade de desativar o mecanismo de gerenciamento de regras de eventos (`enEventDrivenWorkloadAutomation`) que é instalado por padrão com o produto. Consulte o *Guia de Administração* para obter uma lista de opções globais.
composer	Executar a modelagem e o gerenciamento de tarefas das definições de regras de eventos como incluir, criar, excluir, exibir, extrair, listar, bloquear, modificar, novo, imprimir, desbloquear, validar. As regras de eventos são definidas em XML. Consultar o banco de dados relacional do Tivoli Workload Scheduler para: • definições de regras de eventos filtradas por: – regra, evento e propriedades da ação – tarefas e fluxos de tarefas envolvidos com a ação da regra • instâncias de regras de eventos, execução de ações e registros de log de mensagens Consulte "Definição de Regra de Evento" na página 260 para saber como definir regras de eventos. Consulte Capítulo 9, "Gerenciando Objetos no Banco de Dados - Composer", na página 277 para obter a referência de comando.
Dynamic Workload Console	Possuir uma interface gráfica com o usuário para: • Modelar e gerenciar definições de regras de eventos (procurar, criar, excluir, modificar, consultar, desbloquear) • Consultar o banco de dados relacional do Tivoli Workload Scheduler para: – definições de regras de eventos filtradas por: - regra, evento e propriedades da ação - tarefas e fluxos de tarefas envolvidos com a ação da regra – instâncias de regras de eventos, execução de ações e registros de log de mensagens • Gerenciar o servidor de processamento de eventos e mecanismos de monitoramento, conforme descrito nas tabelas Tabela 17 na página 123 e Tabela 18 na página 124 Consulte a documentação do Dynamic Workload Console: Guia do Usuário do Console do Dynamic Workload, seção Criando uma Regra de Evento. Guia do Usuário do Console do Dynamic Workload, seção Tarefas de Gerenciamento de Eventos.
conman	Gerenciar os dispositivos de monitoramento, isto é, o servidor de processamento de eventos e os mecanismos de monitoramento, conforme descrito nas tabelas Tabela 17 na página 123 e Tabela 18 na página 124. Consulte Capítulo 11, "Gerenciando Objetos no Plano - conman", na página 339 para obter a referência de comando.
comandos utilitários	Criar definições de evento customizado e enviar manualmente eventos customizados para o servidor de processamento de eventos. Consulte "evtdef" na página 506 e "sendevent" na página 524 para obter detalhes sobre esses comandos.
planman	Implementar manualmente regras novas e alteradas. Consulte "Regras de Implementação" na página 90 para obter detalhes.
Arquivo de Segurança	Configurar autorizações de segurança para gerenciar regras de eventos, eventos, ações e suas respectivas instâncias. Consulte o *Tivoli Workload Scheduler Administration Guide* para obter referência sobre como configurar o arquivo de segurança do Tivoli Workload Scheduler.

Importante: Se usar um firewall de segurança, certifique-se de que as portas definidas na opção global `eventProcessorEIFPort` e na opção local `nm port` em cada agente estejam abertas para conexões de entrada e de saída.

Definindo Regras de Eventos

Quando você definir uma regra de evento, especifique um ou mais eventos, uma regra de correlação e uma ou mais ações. Para definir as regras de eventos, é possível utilizar:

- A linha de comandos do `composer`
- O Dynamic Workload Console
- Um conjunto de APIs descritas em um documento separado

No `composer`, edite as regras com um editor XML da sua escolha (de preferência, mas não obrigatoriamente), já que as definições de regra de evento são feitas utilizando a sintaxe XML.

A explicação de como você usa o `composer` para definir regras de evento está em "Definição de Regra de Evento" na página 260, enquanto a explicação de como você usa o Dynamic Workload Console pode ser localizada em: Guia do Usuário do Console do Dynamic Workload, seção Criando uma Regra de Evento.

As definições de regras de eventos são salvas no banco de dados do Tivoli Workload Scheduler como todos os outros objetos de planejamento. Você pode salvá-las como:

Rascunho
A regra é salva no banco de dados, mas ainda não está pronta para ser implementada e ativada.

Este estado é determinado pelo atributo `isDraft=yes`.

Não-rascunho
Esta regra está sendo implementada ou está pronta para ser implementada no ambiente de planejamento.

Este estado é determinado pelo atributo `isDraft=no`.

As regras não rascunho estão prontas para ser ativadas. O construtor de regras calcula o status de cada regra. O status pode ser:

- ativo
- não ativo
- atualização pendente
- erro de atualização
- ativação pendente
- erro de ativação
- desativação pendente
- erro de desativação

O planejador varre periodicamente (a cada cinco minutos ou de acordo com um tempo configurado na opção de configuração global `deploymentFrequency`) o banco de dados em busca de regras não rascunho e cria arquivos de configuração de regra para implementação. As novas configurações de monitoramento são transferidas por download para os agentes (cada agente obtém seu próprio arquivo de configuração contendo estritamente as regras que deve executar) somente se tiver havido alterações desde os arquivos de configuração anteriores.

Como recurso adicional, um comando `planman deploy` está disponível para implementar regras manualmente a qualquer momento.

O tempo necessário para implementação aumenta proporcionalmente com o número de regras ativas para implementação. Se for necessário implementar manualmente um grande número de regras novas ou alteradas, e você estiver preocupado em manter o tempo de implementação baixo, execute `planman deploy -scratch` para reduzir o tempo de implementação.

É possível implementar e desimplementar regras conforme necessário configurando o atributo `isDraft` para `no` ou para `yes` com o `composer` ou com o Dynamic Workload Console.

Com base em suas características, as regras são classificadas como:

filter A regra é ativada na detecção de um único evento específico.

sequence
A regra é ativada quando um grupo ordenado de eventos é detectado ou falha em concluir em um intervalo de tempo específico.

set A regra é ativada quando um grupo desordenado de eventos é detectado ou falha em concluir em um intervalo de tempo específico.

As regras de filtragem são baseadas na detecção de um evento como uma tarefa em atraso, uma estação de trabalho do Tivoli Workload Scheduler sendo desativada, um arquivo sendo modificado, um fluxo de tarefas concluindo sua execução com êxito e assim por diante.

As regras set and sequence são baseadas na detecção de mais eventos. Opcionalmente, elas podem se basear em uma condição de tempo limite. Uma regra expira quando o(s) primeiro(s) evento(s) de uma seqüência ou parte dos eventos de um conjunto são recebidos, mas nem todos os eventos são recebidos dentro de um intervalo de tempo especificado, contado a partir do recebimento do primeiro evento.

As definições de regras podem incluir atributos que especificam um período de validade e um espaço de tempo de atividade dentro de cada dia da validade. Se você não especificar estes atributos, a regra será ativa perpetuamente o tempo todo, quando ela é implementada e até ela ser alterada de volta para o status de rascunho ou excluída do banco de dados.

Você pode utilizar a substituição de variável. Isso significa que, ao definir parâmetros de ação, é possível utilizar atributos de ocorrência de eventos que acionam a regra de evento em um destes dois formatos:

- `${event.property}`

  Substitui o valor no estado em que se encontra. Isso é útil para transmitir as informações para uma ação que funciona programaticamente com elas, por exemplo, o `schedTime` de um fluxo de tarefas.
- `%{event.property}`

  Substitui o valor formatado em formato legível a humanos. Isso é útil para passar as informações para uma ação que encaminha essas informações a um usuário, por exemplo para formatar o `schedTime` de um fluxo de tarefas no corpo de um e-mail.

Em que:

evento
É o nome configurado para o acionamento de `eventCondition`.

propriedade

É o nome de `attributeFilter` no predicado de filtragem da condição do evento de acionamento. O valor obtido pelo filtro do atributo quando a regra é acionada é substituído como um valor de parâmetro na definição da ação antes que ela seja executada.

Observe que você também pode utilizar a substituição de variável quando nenhum `attributeFilter` foi especificado para um atributo e quando o atributo for de leitura.

É possível ver o uso da substituição de variável em algumas das seguintes amostras, em que os valores do filtro de atributos são substituídos no assunto e no corpo do e-mail.

Exemplos de Regras de Eventos

Seguem exemplos de definições de regras de eventos que se aplicam aos cenários descritos em "Cenários Simples de Regras de Eventos" na página 120.

Definição de Regra de Evento para Cenário No. 1

Quando alguma das tarefas `job123` finaliza com erro e gera a seguinte mensagem de erro:

```
AWSBHT001E The job
"MYWORKSTATION#JOBS.JOB1234" in file "ls" has failed with
the error: AWSBDW009E The following operating system error occurred
retrieving
the password structure for either the logon user...
```

enviar um e-mail ao operador `john.smith@mycorp.com`. O assunto do e-mail inclui os nomes da instância da tarefa e da estação de trabalho associada.

A regra de evento é válida de 1° de Dezembro a 31 de Dezembro no espaço de tempo 12h-16h, horário do leste.

```
<?xml version="1.0"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules"
      xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/
                          event-management/rules/EventRules.xsd">
   <eventRule name="scenario1_rule" ruleType="filter"
isDraft="no">
      <description>This is the definition for
scenario1</description>
      <timeZone>America/Indianapolis</timeZone>
      <validity from="2010-12-01" to="2010-12-31" />
      <activeTime start="12:00:00" end="16:00:00" />
      <eventCondition name="event1"
                      eventProvider="TWSObjectsMonitor"
                      eventType="JobStatusChanged">
         <filteringPredicate>
            <attributeFilter name="JobStreamWorkstation" operator="eq">
               <value>*</value>
            </attributeFilter>
            <attributeFilter name="JobStreamName" operator="eq">
               <value>*</value>
            </attributeFilter>
            <attributeFilter name="JobName" operator="eq">
               <value>job123*</value>
            </attributeFilter>
            <attributeFilter name="Status" operator="eq">
               <value>Error</value>
            </attributeFilter>
            <attributeFilter name="ErrorMessage" operator="eq">
               <value>*AWSBDW009E*</value>
            </attributeFilter>
         </filteringPredicate>
      </eventCondition>
      <action actionProvider="MailSender"
              actionType="SendMail"
              responseType="onDetection">
         <description>Send email to John Smith including names of job
                      and associated workstation</description>
         <parameter name="To">
            <value>john.smith@mycorp.com</value>
         </parameter>
         <parameter name="Subject">
            <value>Job %{event1.JobName} on agent %{event1.Workstation}
                   ended in error</value>
```

```
        </parameter>
      </action>
   </eventRule>
</eventRuleSet>
```

Importante: A mensagem de erro que explica porque uma tarefa finaliza com erro pode ser localizada no arquivo de log TWSMERGE. Nesse cenário, o arquivo de log TWSMERGE contém a seguinte instrução:

```
BATCHMAN:+
BATCHMAN:+ AWSBHT001E The job "MYWORKSTATION#JOBS.JOB1234" in file
"ls"
has failed with the error: AWSBDW009E The following operating system
error occurred retrieving the password structure for either the logon
user, or the user who owns a file or external dependency
BATCHMAN:+
```

em que a mensagem de erro é tudo o que segue na sequência:

```
has failed with the error:
```

Definição de Regra de Evento para Cenário No. 2

Se a estação de trabalho`CPU1` for desconectada e não se conectar novamente dentro de 1 hora, envia um e-mail de notificação para `chuck.derry@mycorp.com`.

```
<?xml version="1.0"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules"
      xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/
                          event-management/rules/EventRules.xsd">
   <eventRule name="scenario2_rule" ruleType="filter" isDraft="no">
      <description>This is the definition for scenario2</description>
      <timeZone>America/Anchorage</timeZone>
      <timeInterval amount="1" unit="hours" />
      <eventCondition name="WSevent"
                      eventProvider="TWSObjectsMonitor"
                      eventType="ChildWorkstationLinkChanged">
         <filteringPredicate>
            <attributeFilter name="Workstation" operator="eq">
               <value>CPU1</value>
            </attributeFilter>
            <attributeFilter name="LinkStatus" operator="eq">
               <value>Unlinked</value>
            </attributeFilter>
         </filteringPredicate>
      </eventCondition>
      <action actionProvider="MailSender"
              actionType="SendMail"
              responseType="onDetection">
         <description>Send email to Chuck Derry with name of
                      unlinked workstation</description>
         <parameter name="To">
             <value>chuck.derry@mycorp.com</value>
         </parameter>
         <parameter name="Subject">
            <value>Agent CPU1 has been unlinked for at least 10 minutes</value>
         </parameter>
         <parameter name="Body">
             <value>The cause seems to be: %{WSevent.UnlinkReason}</value>
         </parameter>
      </action>
   </eventRule>
</eventRuleSet>
```

Definição de Regra de Evento para Cenário No. 3

Quando o arquivo `daytransac` for criado no diretório `SFoperation` na estação de trabalho`system1` e as modificações no arquivo forem finalizadas, envie o fluxo de tarefas `calmonthlyrev`.

A regra de evento é válida durante o ano no espaço de tempo das 18h às 22h no horário do leste.

```
<?xml version="1.0"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules"
      xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/
                          event-management/rules/EventRules.xsd">
   <eventRule name="scenario3_rule"
              ruleType="filter"
              isDraft="no">
      <description>This is the definition for scenario3</description>
      <timeZone>America/Louisville</timeZone>
      <validity from="2007-01-01" to="2007-12-31" />
      <activeTime start="18:00:00" end="22:00:00" />
      <eventCondition eventProvider="FileMonitor"
                      eventType="ModificationCompleted">
         <filteringPredicate>
            <attributeFilter name="FileName" operator="eq">
```

```
            <value>daytransac</value>
         </attributeFilter>
         <attributeFilter name="Workstation" operator="eq">
            <value>EVIAN1</value>
         </attributeFilter>
      </filteringPredicate>
   </eventCondition>
   <action actionProvider="TWSAction"
           actionType="sbs"
           responseType="onDetection">
      <description>Submit the calmonthlyrev job stream.</description>
      <parameter name="JobStreamName">
         <value>calmonthlyrev</value>
      </parameter>
      <parameter name="JobStreamWorkstationName">
         <value>act5cpu</value>
      </parameter>
   </action>
 </eventRule>
</eventRuleSet>
```

Definição de Regra de Evento para Cenário No. 4

Quando o eventojob-x=exec e o eventojob-x=succ/abend forem recebidos em 500 segundos, o planejador deve responder Sim aoprompt-1 e inicia o fluxo de tarefas jobstream-z, caso contrário ele deve enviar um e-mail para twsoper@mycompany.com alertando que a tarefa está atrasada.

```
<?xml version="1.0"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules"
      xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/
                          event-management/rules/EventRules.xsd">
   <eventRule name="scenario4_rule"
              ruleType="sequence"
              isDraft="yes">
      <description>This is the definition for
scenario4</description>
      <timeZone>America/Buenos_Aires</timeZone>
      <timeInterval amount="500" unit="seconds" />
      <eventCondition eventProvider="TWSObjectsMonitor"
                      eventType="JobStatusChanged">
         <filteringPredicate>
            <attributeFilter name="JobName" operator="eq">
               <value>job-x</value>
            </attributeFilter>
            <attributeFilter name="InternalStatus" operator="eq">
               <value>EXEC</value>
            </attributeFilter>
         </filteringPredicate>
      </eventCondition>
      <eventCondition eventProvider="TWSObjectsMonitor"
                      eventType="JobStatusChanged">
         <filteringPredicate>
            <attributeFilter name="JobName" operator="eq">
                <value>job-x</value>
            </attributeFilter>
            <attributeFilter name="InternalStatus" operator="eq">
                <value>ABEND</value>
                <value>SUCC</value>
            </attributeFilter>
         </filteringPredicate>
      </eventCondition>
      <action actionProvider="MailSender"
              actionType="SendMail"
              responseType="onTimeOut">
         <description>Send email to operator saying that job-x
                      is late</description>
         <parameter name="To">
            <value>twsoper@mycorp.com</value>
         </parameter>
         <parameter name="Subject">
            <value>Job-x is late by at least 5 minutes</value>
         </parameter>
      </action>
      <action actionProvider="TWSAction"
              actionType="Reply"
              responseType="onDetection">
         <description>Reply Yes to prompt-1</description>
         <parameter name="PromptName">
            <value>prompt-1</value>
         </parameter>
         <parameter name="PromptAnswer">
            <value>Yes</value>
         </parameter>
      </action>
      <action actionProvider="TWSAction"
              actionType="sbs"
              responseType="onDetection">
         <description>Submit jobstream-z</description>
         <parameter name="JobStreamName">
            <value>jobstream-z</value>
         </parameter>
         <parameter name="JobStreamWorkstationName">
            <value>act23cpu</value>
         </parameter>
      </action>
   </eventRule>
</eventRuleSet>
```

Definição de Regra de Evento para o Cenário Nº 5

Monitora o status de processos do Tivoli Workload Scheduler listados em ProcessName e executa o script em lote RUNCMDFM.BAT localizado em E:\production\eventRules.

A palavra-chave TWSPATH indica o caminho completo em que a instância do Tivoli Workload Scheduler monitorada está instalada, incluindo o sufixo /TWS.

Em sistemas operacionais Windows, a regra de evento é acionada toda vez que o agente é interrompido usando o comando ShutDownLwa e toda vez que o agente é parado manualmente. Em sistemas operacionais UNIX, a regra de evento é acionada quando o processo é interrompido manualmente, embora ela não seja acionada pelo comando ShutDownLwa.

Se ProcessName=agent for especificado, o componente do agente é monitorado, embora o processo JobManager do TWS não seja monitorado.

```
<?xml version="1.0"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules"
      xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/
                          event-management/rules/EventRules.xsd">
    <eventRule name="scenario5rule" ruleType="filter"
isDraft="no">
        <eventCondition name="twsProcMonEvt1"
        eventProvider="TWSApplicationMonitor"
        eventType="TWSProcessMonitor">
            <scope>
                AGENT, NETMAN DOWN ON WIN86MAS
            </scope>
            <filteringPredicate>
                <attributeFilter name="ProcessName"
operator="eq">
                 <value>agent</value>
                 <value>appservman</value>
                 <value>batchman</value>
                 <value>jobman</value>
                 <value>mailman</value>
                 <value>monman</value>
                 <value>netman</value>
       </attributeFilter>
                <attributeFilter name="TWSPath" operator="eq">
                   <value>E:\Program
Files\IBM\TWA\TWS</value>
                </attributeFilter>
                <attributeFilter name="Workstation" operator="eq">
                    <value>win86mas</value>
                </attributeFilter>
                <attributeFilter name="SampleInterval" operator="eq">
                    <value>5</value>
                </attributeFilter>

            </filteringPredicate>
        </eventCondition>
        <action actionProvider="GenericActionPlugin"
actionType="RunCommand"
    responseType="onDetection">
            <scope>
                RUNCMDFM.BAT
            </scope>
            <parameter name="Command">
                <value>runCmdFM.bat</value>
            </parameter>
            <parameter name="WorkingDir">
                <value>E:\production\eventRules</value>
            </parameter>
        </action>
    </eventRule>
</eventRuleSet>
```

Definição de Regra de Evento para Cenário Nº7

Monitora o status do arquivo Symphony na estação de trabalho*IT041866-9088* e registra a ocorrência de um registro Symphony corrompido no banco de dados de logs de auditoria interna.

Em cada estação de trabalho, o processo do Batchman monitora o arquivo Symphony. Quando ele detecta um registro corrompido, ele envia o evento de dano para a fila de mensagens do processo Monman e depois para o Processador de Eventos na estação de trabalho Principal.

A regra de evento é acionada sempre que o processo do Batchman localiza um registro de dependência corrompido na estação de trabalho especificada na definição da regra de evento.

Se você configurar o valor da estação de trabalho para IT041866-9088, o arquivo Symphony nessa estação de trabalho será monitorado, e a regra de evento será acionada quando o processo do Batchman nessa estação de trabalho detectar um registro corrompido no arquivo Symphony.

A ocorrência do registro corrompido é gravada no arquivo de auditoria do criador de logs Messagge.

```
?xml
version="1.0"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules"
xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules
http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules/EventRules.xsd">
<eventRule name="TEST1" ruleType="filter" isDraft="no">
<eventCondition name="productAlertEvt1" eventProvider="TWSObjectsMonitor"
eventType="ProductAlert">
<scope>
IT041866-9088
</scope>
<filteringPredicate>
<attributeFilter name="Workstation" operator="eq">
<value>IT041866-9088</value>
</attributeFilter>

</filteringPredicate>
</eventCondition>
<action actionProvider="MessageLogger" actionType="MSGLOG"
responseType="onDetection">
<scope>
OBJECT=%{PRODUCTALERTEVT1.WORKSTATION} MESSAGE=%{PRODUCTALERTEVT1.WORKSTATION}
corruption
</scope>
<parameter name="ObjectKey">
<value>%{productAlertEvt1.Workstation}</value>
</parameter>
<parameter name="Message">
<value>%{productAlertEvt1.Workstation} corruption</value>
</parameter>
<parameter name="Severity">
<value>Info</value>
</parameter>
</action>
</eventRule>
</eventRuleSet>
```

Notas sobre a operação de regras

Seguem informações críticas sobre o comportamento da regra de evento que podem ajudá-lo a entender a razão de resultados inesperados:

Notas sobre status da regra

- Dependendo das datas de validade `from` e `to` correspondentes, o status de qualquer regra é alterado da seguinte forma na implementação:
  - Se você criar uma regra com as datas de validade `from` e `to` já expiradas, a regra será salva no estado`activation pending`. Quando a regra é implementada, ela permanece no status `activation pending`.
  - Se você configurar o campo de validade `to` para uma data futura, a regra será implementada no estado`active`. Se você reconfigurá-la para uma data no passado, a regra será reimplementada no estado`no active`.
- Os horários de atividade da regra (`start` e `end`) não afetam o seu status. Enquanto uma regra está dentro das datas de validade corretas, a regra permanece no estado`active`, independentemente de ela estar ou não dentro de seus horários de atividade definidos. Se o planejador receber os eventos definidos de uma regra fora de seu horário de atividade, os eventos serão descartados, mas a regra permanecerá no status `active`.

Falta de persistência no status da instância de regra

Se o processador de eventos for parado ou travar, o status das instâncias de regra apenas parcialmente correspondidas é perdido.

Regras set and sequence repetitivas

Em geral, cada regra ativa possui uma, e apenas uma, cópia executada no servidor de processamento de eventos. As regras `Set` e `sequence` utilizam o mecanismo explicado no seguinte exemplo:

1. Você define uma regra `sequence` com dois eventos, A e B.
2. Quando o primeiro evento que corresponde à seqüência ocorre (evento A), ele ativa a regra e aguarda o segundo evento (evento B).

   Quando a regra estiver ativa, qualquer evento adicional do evento A que possa chegar será ignorado. Nenhuma regra adicional será criada para quaisquer eventos do evento A recém-detectado, uma vez que a regra está aguardando pelo evento B.
3. Quando o evento B ocorrer, a regra será concluída e reconfigurada, aguardando que o evento A ocorra novamente.

O mecanismo das regras `set` e `sequence` ocorre de forma que qualquer ocorrência adicional de um evento já detectado seja ignorada e descartada se os outros eventos definidos não tiverem chegado.

É possível evitar esse problema utilizando atributos de correlação. Utilizar um ou mais atributos de correlação é um método de direcionar uma regra para criar uma cópia da regra separada quando outro evento A chega.

Tipos de regras set e sequence com períodos de tempo de atividade com menos de 24 horas

Ocorrências de regras `set` ou `sequence` que foram definidas para estarem ativas por algumas poucas horas de cada dia não são limpas quando o período de atividade em cada dia expira, se apenas parte dos eventos chega. Elas permanecem em um estado inativo, aguardando para receber os eventos restantes nos dias seguintes.

Por exemplo, você define uma regra `set` que inclui dois eventos. A regra é válida de 1° de Janeiro a 10 de Janeiro e está ativa diariamente das 6h às 10h.

Se em 1° de Janeiro o primeiro evento for recebido às 8h, a regra aguardará que o segundo evento ocorra e permanecerá *ativa* além das 10h se o evento não for detectado.

Se o segundo evento for recebido às 11h (que está fora do espaço de tempo da atividade), ele será descartado, mas a regra se manterá *ativa*. Se o segundo evento for recebido novamente às 7h de 2 de Janeiro, a regra será acionada e as ações serão implementadas.

Se não desejar que a regra seja *adiada* para o dia seguinte, você deve limpá-la.

Elementos de Regra Acionados

Toda vez que as condições de evento listadas em uma regra de evento implementada são correspondidas, ou atingem o tempo limite, uma instância de regra de evento é criada. Uma instância de regra de evento inclui informações como o nome da regra de evento, a data e hora em que ela foi correspondida e a lista de instâncias de ação, com os respectivos status, que foram executadas como resultado das condições de evento correspondentes. O objeto`RuleInstance` é utilizado para coletar informações sobre as regras acionadas em um log histórico de instâncias da regra.

As ações executadas por uma regra acionada são coletadas em um log de histórico de execuções da ação. As informações fornecidas incluem execuções de ação que foram concluídas com erros ou avisos, em oposição às bem-sucedidas. Se pelo menos uma ação encerrar com erro, toda a instância de regra será relatada com erro. Como parte das informações rastreadas nas execuções de ação, os campos de regra também são mantidos e as consultas podem ser executadas para procurar execuções de ação com base nestes campos (o identificador da instância da regra também está disponível).

Definindo Eventos Customizados

Além dos tipos e classes de eventos já definidos (conhecidos como provedores) listados em detalhes em "Provedores e Definições de Eventos" na página 651, o Tivoli Workload Scheduler fornece o modelo de um provedor de eventos genéricos denominado `GenericEventPlugIn` que os programadores com habilidades específicas de programação de aplicativos e XML podem modificar a fim de definir tipos de eventos customizados que podem ser úteis para a organização.

As ferramentas fornecidas para definir tipos de eventos customizados são:
- O provedor de eventos `GenericEventPlugIn` em XML
- O comando utilitário evtdef com o qual um programador pode fazer o download do provedor de eventos `GenericEventPlugIn` como um arquivo local a fim de definir os eventos customizados
- Os arquivos XSD (XML Schema Definition) necessários para validar o provedor de eventos genéricos modificado. Eles também contêm orientações online para auxiliar na tarefa de programação.
- O comando utilitário sendevent com o qual os eventos customizados podem ser enviados ao servidor de processamento de eventos para acionar regras a partir de qualquer agente ou qualquer estação de trabalho, simplesmente executando o cliente da linha de comandos remoto do Tivoli Workload Scheduler.

Este é o fluxo para definir e utilizar eventos customizados:

1. Com o comando evtdef, o programador:
   a. Faz download do provedor de eventos genéricos como um arquivo local.
   b. Segue as definições do esquema para incluir tipos de evento customizado e para definir suas propriedades e atributos no arquivo com um editor XML.
   c. Faz upload do arquivo local como o provedor de eventos genéricos modificado contendo as novas definições de tipo de evento customizado. O provedor de eventos genéricos modificado é salvo em um arquivo XML no gerenciador de domínio principal.
2. O construtor de regras, ou o administrador, define, com o criador ou o Dynamic Workload Console, as regras de eventos que devem ser acionadas por esses eventos customizados, especificando:
   - O provedor de eventos genéricos como o provedor de eventos
   - Os tipos de evento customizado como os tipos de evento
   - As propriedades do tipo de evento customizado (ou atributos) definidas para os eventos customizados no provedor de eventos genéricos com os valores específicos que irão acionar as regras.
3. Implementar as regras.
4. Quando a ocorrência de um evento customizado acontece, ela pode ser enviada ao servidor de processamento de eventos de uma das seguintes maneiras:
   - Por meio do comando sendevent, executado a partir de um script ou da linha de comandos

- Por meio de outro aplicativo, como o Tivoli Enterprise Console ou o Tivoli Monitoring

Assim que o evento é recebido pelo servidor de processamento de eventos, ele aciona a regra.

Capítulo 8. Definindo Objetos no Banco de Dados

O Tivoli Workload Scheduler é baseado em um conjunto de definições de objetos que são utilizadas para mapear seu ambiente de planejamento no banco de dados. Os objetos de planejamento são:

- calendários
- domínios
- regras de eventos
- tarefas
- fluxos de tarefas
- prompts
- parâmetros
- recursos físicos
- grupos de ciclos de execução
- tabelas de variáveis
- aplicativos de carga de trabalho
- estações de trabalho
- classes de estações de trabalho
- usuários

Definindo Objetos de Planejamento

Os objetos de planejamento são gerenciados com o programa da linha de comandos **composer** e são armazenados no banco de dados do Tivoli Workload Scheduler. O programa de linha de comandos **composer** pode ser instalado e utilizado em qualquer máquina conectada através de TCP/IP ao sistema onde ogerenciador de domínio principal está instalado. Não requer a instalação de uma estação de trabalho do Tivoli Workload Scheduler como um pré-requisito. Comunica-se através de HTTP/HTTPS com o gerenciador de domínio principal onde o banco de dados DB2 está instalado. A configuração da comunicação HTTP/HTTPS e a verificação de autenticação são gerenciadas pela infra-estrutura do WebSphere Application Server. O programa **composer** utiliza arquivos de edição para atualizar o banco de dados de planejamento. O formato do arquivo de edição utilizado para definir um objeto específico é descrito posteriormente neste capítulo. Por exemplo, para criar um novo objeto, digite sua definição em um arquivo de edição e, em seguida, utilize o**composer** para incluí-la no banco de dados especificando o arquivo de edição que contém a definição. O programa da linha de comandos **composer** verifica se a sintaxe está correta dentro do arquivo de edição e, se correta, transforma a definição de objeto em linguagem XML e, em seguida, envia o pedido através de HTTP/HTTPS para o gerenciador de domínio principal.

No gerenciador de domínio principal, a definição XML é analisada. São realizadas verificações de semântica e de integridade e, em seguida, a atualização é armazenada no banco de dados.

Com esta versão do produto, todas as entradas são gerenciadas individualmente. Outro recurso introduzido com esta nova versão é o mecanismo de bloqueio de objetos. Os objetos de planejamento definidos no banco de dados são bloqueados enquanto são acessados por um usuário para evitar acessos simultâneos. Isto significa que somente o usuário que bloqueia o objeto possui permissão de

© Copyright IBM Corp. 1999, 2014

gravação para esse objeto e outros usuários possuem acesso de leitura para esse objeto. Para obter informações adicionais, consulte “lock” na página 310 e “unlock” na página 326.

É possível utilizar palavras-chave curtas e longas ao emitir comandos a partir do **composer**, conforme Tabela 20 mostra. As primeiras duas colunas na lista à esquerda são os formatos de palavra-chave longa e curta suportados atualmente pelo Tivoli Workload Scheduler. A coluna mais à direita lista os formatos compatíveis com versões anteriores que você deseja utilizar se sua rede incluir instalações pré-versão 8.3.

Tabela 20. Lista de Palavras-chave de Objetos de Planejamento Suportadas

Palavras-chave Longas	Palavras-chave Curtas	Palavras-chave compatíveis com instalações anteriores à versão 8.3
calendário	cal	calendários
domínio	dom	cpu
eventrule	erule | er	—
jobdefinition	jd	tarefas
jobstream	js	sched
parameter	parm	parms
prompt	prom	prompts
recurso	res	recursos físicos
usuário	usuário	usuários
variabletable	vt	—
estação de trabalho	ws	cpu
workstationclass	wscl	cpu

Nota: A palavra-chave **cpu** é mantida para representar domínios, estações de trabalho e classes de estação de trabalho para compatibilidade com versões anteriores.

O programa **composer** não emite avisos específicos se as palavras-chave de linguagem de planejamento forem utilizadas como nomes de objetos de planejamento. No entanto, a utilização dessas palavras-chave pode resultar em erros, portanto, evite a utilização das palavras-chave listadas na Tabela 21 ao definir tarefas e fluxos de tarefas:

Tabela 21. Lista de Palavras Reservadas ao Definir Tarefas e Fluxos de Job

Lista de palavras reservadas:				
abendprompt	after	as	às	autodocoff
autodocon	canc	carryforward	confirmed	continuar
dateval	day(s)	day_of_week	prazo final	descrição
docommand	draft	end	a cada	everyday
except	extraneous	fdignore	fdnext	fdprev
nome do arquivo	follows	dias livres	from	ir
hi	i18n_id	i18n_priority	interactive	isdefault
isuserjob	jobfilename	tarefas	keyjob	keysched
limite	matching	members	needs	nextjob
notempty	número	em	onuntil	op
opens	order	previous	prioridade	prompt
qualifier	rccondsucc	recovery	relative	request
executar novamente	runcycle	sa	sameday	schedtime
planejamento	scriptname	parar	streamlogon	su
tasktype	timezone	to	token_in	until
validfrom	validto	vartable	vt	weekday(s)
workday(s)				

Evite utilizar as palavras-chave listadas em Tabela 22 na página 139 ao definir estações de trabalho, classes de estação de trabalho e domínios:

Tabela 22. Lista de Palavras Reservadas ao Definir Estações de Trabalho

Lista de palavras reservadas:				
access	AIX	agent_type	autolink	behindfirewall
comando	cpuclass	cpuname	descrição	domínio
ativado	end	extraneous	for	forçar
fta	fullstatus	host	hpux	ibm i
ignorar	isdefault	linkto	maestro	manager
principal	members	mpeix	mpev	mpexl
mpix	nó	número	off	em
os	other	parent	posix	servidor
secureaddr	securitylevel	tcpaddr	timezone	tipo
tz	tzid	UNIX	utilizando	vartable
wnt				

Evite usar as palavras-chaves listadas em Tabela 23 ao definir usuários:

Tabela 23. Lista de Palavras Reservadas ao Definir Usuários

Lista de palavras reservadas:		
username	password	end

Utilizando Modelos de Definição de Objeto

Os modelos de definição de objeto de planejamento estão disponíveis para uso no diretório *TWS_home*\templates. É possível utilizar os modelos como um ponto inicial quando você define objetos de planejamento.

Observe que as datas nos modelos estão no formato expresso na opção local `date format`.

Definição de estação de trabalho

Em uma rede do Tivoli Workload Scheduler, uma estação de trabalho é um objeto de planejamento que executa tarefas. Você cria e gerencia a estação de trabalho do objeto de planejamento no banco de dados do Tivoli Workload Scheduler utilizando uma definição da estação de trabalho. Uma definição da estação de trabalho é necessária para cada objeto que executa tarefas. Geralmente, uma definição de estação de trabalho é usada para representar uma estação de trabalho física, mas, no caso de agentes estendidos, por exemplo, representa uma definição lógica que deve ser hospedada por uma estação de trabalho física.

É possível incluir diversas definições de estação de trabalho no mesmo arquivo de texto juntamente com as definições de classe de estação de trabalho e as definições de domínio. Uma definição de estação de trabalho tem a seguinte sintaxe:

Sintaxe

cpuname *workstation* [**description**
"*description*"]
[**vartable** *table_name*]
os *os-type*
[**node** *hostname*] [**tcpaddr** *port*]
[**secureaddr** *port*] [**timezone**|**tz** *tzname*]
[**domain** *domainname*]
[**for maestro** [**host** *host-workstation*
[**access** *method* | **agentID** *agentID*]]
[**type fta** | **s-agent** | **x-agent** | **manager** | **broker** | **agent** | **rem-eng** | **pool** | **d-pool**]

[ignore]
[autolink on | off]
[behindfirewall on | off]
[securitylevel enabled | on | force]
[fullstatus on | off]
[server *serverid*]]
[protocol *http | htpps*]
[members [*workstation*] [...]]
[requirements *jsdl_definition*]]
end

[cpuname ...]

[cpuclass ...]

[domain ...]

Argumentos

Tabela 24. Configurações de Atributos para Tipos de Estação de Trabalho de Gerenciamento

Attributes	Gerenciador de domínio principal	Gerenciador de domínio	Gerenciador de Domínio de Backup
cpuname	O nome da estação de trabalho.		
description	Uma descrição para a estação de trabalho colocada entre aspas duplas. Este atributo é opcional.		
vartable	O nome de uma tabela de variável associada à estação de trabalho. Variáveis usadas com a estação de trabalho são definidas nesta tabela. Este atributo é opcional.		

os	O sistema operacional instalado no sistema. Especifique um dos seguintes valores:
	
	`UNIX`
	`WNT`
	`OTHER`
	`IBM_i`		

nó	O nome do host ou o endereço IP do sistema.		
tcpaddr	O valor designado a *nm port* no arquivo `localopts`. Para diversas estações de trabalho em um sistema, insira um número de porta não usado. O valor padrão é 31111.		
secureaddr	O valor designado a *nm ssl port* no arquivo `localopts`. Especifique-o se **securitylevel** estiver configurado para `on`, `force` ou `enabled`.		
timezone | **tz**	O fuso horário no qual o sistema está localizado. É recomendável que o valor corresponda ao valor configurado no sistema operacional.		
domínio	`MASTERDM`	O nome do domínio gerenciado.	
host	Não aplicável		
access	Não aplicável		
type	`manager`		`fta`
ignorar	Use esse atributo se não quiser que essa estação de trabalho seja incluída no próximo plano de produção.		

autolink	Indica se um link entre estações de trabalho é automaticamente aberto na inicialização. Especifique um dos seguintes valores:
	
	`ON`
	`OFF`
	
	Esse é um atributo opcional. O valor padrão é `ON`.		

behindfirewall	Essa configuração é ignorada.	Indica se há um firewall entre a estação de trabalho e o gerenciador de domínio principal. Especifique um dos seguintes valores:
		
		`ON`
		`OFF`
		
		O valor padrão é `OFF`.	

securitylevel	O tipo de autenticação SSL a usar:
	
	`enabled`
	`on`
	`forçar`		

fullstatus	`ON`		
servidor	Não aplicável		Essa configuração é ignorada.
protocol	Não aplicável		

Tabela 24. Configurações de Atributos para Tipos de Estação de Trabalho de Gerenciamento (continuação)

Attributes	Gerenciador de domínio principal	Gerenciador de domínio	Gerenciador de Domínio de Backup
members	Não aplicável		
requisitos	Não aplicável		

O Tabela 25 descreve os valores que você configura para cada atributo para tipos de estação de trabalho de destino. Após a tabela, há detalhes adicionais sobre cada atributo.

Tabela 25. Configurações de Atributos para Tipos de Estação de Trabalho de Destino

Atributo	Agente tolerante a falhas e agente padrão	Estação de trabalho do Workload Broker	Agente Estendido	Agent	Estação de trabalho de mecanismo remoto	Conjunto	Conjunto dinâmico
cpuname	O nome da estação de trabalho.						
description	Uma descrição para a estação de trabalho colocada entre aspas duplas. Este atributo é opcional.						
vartable	O nome de uma tabela de variável associada à estação de trabalho. Variáveis usadas com a estação de trabalho são definidas nesta tabela. Este atributo é opcional.						

os	O sistema operacional instalado no sistema. Especifique um dos seguintes valores:
	
	`UNIX`
	`WNT`
	`OTHER`
	`IBM_i`
	
	Especifique `OTHER` para sistemas IBM i executando como agentes tolerantes a falhas limitados.
		`OTHER`	O sistema operacional instalado na máquina. Especifique um dos seguintes valores:
			
			`UNIX`
			`WNT`
			`OTHER`
			`IBM_i`
				A configuração desse valor é descoberta no sistema.	O sistema operacional instalado na máquina. Especifique um dos seguintes valores:
					
					`UNIX`
					`WNT`
					`ZOS`
						O sistema operacional instalado na máquina. Especifique um dos seguintes valores:
						
						`UNIX`
						`WNT`
						`OTHER`
						`IBM_i`	

nó	O nome do host ou o endereço IP do sistema.		O nome do host ou o endereço IP do sistema. Especifique `NULL` quando **host** for configurado para `$MASTER` ou ao definir um agente estendido para PeopleSoft, SAP ou Oracle.	O nome do host ou o endereço IP do agente.	O nome do host ou o endereço IP do mecanismo remoto.	Não aplicável	
tcpaddr	O valor designado a *nm port* no arquivo `localopts`. Ao definir diversas estações de trabalho em um sistema, insira um número de porta não usado. O valor padrão é 31111.	O valor designado a *nm port* no arquivo `localopts`. Ao definir diversas estações de trabalho em um sistema, insira um número de porta não usado. O valor padrão é 41114.	Consulte as especificações do método de acesso selecionado.	O número da porta para se comunicar com o agente quando o protocolo é `http`.	O número da porta para se comunicar com o mecanismo remoto quando o protocolo é `http`.	Não aplicável	
secureaddr	O valor designado a *nm ssl port* no arquivo `localopts`. Especifique-o se **securitylevel** estiver configurado para `on`, `force` ou `enabled`.	Não Aplicável	Não Aplicável	O número da porta para se comunicar com o agente quando o protocolo é `https`.	O número da porta para se comunicar com o mecanismo remoto quando o protocolo é `https`.	Não aplicável	
timezone | tz	O fuso horário no qual o sistema está localizado. É recomendável que o valor corresponda ao valor configurado no sistema operacional.		O fuso horário configurado na estação de trabalho especificada no atributo **host**.	O fuso horário configurado no agente.	O fuso horário configurado no mecanismo remoto.	O fuso horário configurado nos agentes do conjunto.	O fuso horário configurado nos agentes do conjunto dinâmico.
domínio	Especifique um domínio existente. O valor padrão para agentes tolerantes a falhas é `MASTERDM`. Essa configuração é obrigatório para agentes padrão.	Especifique um domínio existente. Essa configuração é obrigatória.	Essa configuração é necessária somente se o valor designado para **host** for `$MANAGER`.	Não aplicável			
host	Não Aplicável		A estação de trabalho host. Pode ser configurada para `$MASTER` ou `$MANAGER`.	A estação de trabalho de broker.			
access	Não Aplicável			Selecione o nome do arquivo do método de acesso apropriado.	Não Aplicável		

Tabela 25. Configurações de Atributos para Tipos de Estação de Trabalho de Destino (continuação)

Atributo	Agente tolerante a falhas e agente padrão	Estação de trabalho do Workload Broker	Agente Estendido	Agent	Estação de trabalho de mecanismo remoto	Conjunto	Conjunto dinâmico
agentID				O identificador exclusivo do agent			

type	`fta`
	`s-agent`
	
	O valor padrão é `fta`.
	
	Especifique `fta` para sistemas IBM i executando como agentes tolerantes a falhas limitados.
		`broker`	`agente X`	`agent`	`rem-eng`	`conjunto`	`d-pool`

ignorar	Use esse atributo se não quiser que essa estação de trabalho apareça no próximo plano de produção.						

autolink	Indica se um link entre estações de trabalho é aberto automaticamente na inicialização. Especifique um dos seguintes valores:
	
	`ON`
	`OFF`
	
	Esse é um atributo opcional. O valor padrão é `ON`.
			`OFF`	Não aplicável			

behindfirewall	Indica se há um firewall entre a estação de trabalho e o gerenciador de domínio principal. Especifique um dos seguintes valores:
	
	`ON`
	`OFF`
	
	O valor padrão é `OFF`.
			`OFF`	Não aplicável			

securitylevel	O tipo de autenticação SSL a usar:
	
	`enabled`
	`on`
	`forçar`
	
	Não aplicável para sistemas IBM i executando como agentes tolerantes a falhas limitados.
		Não Aplicável					

fullstatus	Indica se a estação de trabalho é atualizada com relação ao status de processamento de tarefa em seu domínio ou subdomínios. Especifique um dos seguintes valores:
	
	`ON`
	`OFF`
	
	Especifique `OFF` para agentes padrão.
		`OFF`		Não aplicável			

servidor	0-9, A-Z. Quando especificado, requer a criação de processos de um mailman dedicado na estação de trabalho pai.		Não Aplicável				

protocol	Não aplicável			Especifique um dos seguintes valores:
				
				`http`
				`https`
				
				Este atributo é opcional. Quando não especificado, é determinado automaticamente das configurações especificadas para **tcpaddr** e **secureaddr**.
						Não aplicável	

members	Não aplicável					Valor obrigatório	Não aplicável
requisitos	Não aplicável						Valor obrigatório

A lista a seguir fornece detalhes adicionais para os atributos de definição da estação de trabalho:

cpuname *workstation*
Especifica o nome da estação de trabalho. Os nomes de estações de trabalho devem ser exclusivos e não podem ser os mesmos das classes de estações de trabalho.

O nome deve iniciar com uma letra e pode conter caracteres alfanuméricos, traços e sublinhados. Ele pode conter até 16 caracteres.

Não use neste campo nenhuma das palavras reservadas especificadas em Tabela 21 na página 138.

description "*description*"
Fornece uma descrição da estação de trabalho. A descrição pode conter até 120 caracteres alfanuméricos. O texto deve ser colocado entre aspas duplas.

vartable *table_name*
Especifica o nome da tabela de variável que você deseja associar à estação de trabalho. Variáveis usadas com a estação de trabalho são definidas nesta tabela.

O nome deve iniciar com uma letra e pode conter caracteres alfanuméricos, traços e sublinhados. Ela pode conter até 80 caracteres.

os *os_type*
Especifica o sistema operacional da estação de trabalho. Quando usado em definições de estação de trabalho de mecanismo remoto, representa o sistema operacional do mecanismo remoto do Tivoli Workload Scheduler.

Os valores válidos são:

UNIX Para sistemas operacionais suportados em execução em sistemas baseados em UNIX, incluir sistemas LINUX.

WNT Para sistemas operacionais Windows suportados.

OUTRO
Valor obrigatório para: estações de trabalho do Tivoli dynamic workload broker e sistemas IBM i executando como agentes tolerantes a falhas limitados. Valor opcional para outros tipos de estações de trabalho.

ZOS Usado com estações de trabalho de mecanismo remoto definidas para comunicação com o mecanismo remoto do Tivoli Workload Scheduler for z/OS.

IBM_i Para sistemas operacionais IBM i suportados.

Nota: Para uma lista atualizada de sistemas operacionais suportados, consulte o documento *IBM Tivoli Workload Scheduler System Requirements* em http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27041009.

node *hostname*
Especifique o nome do host ou o endereço TCP/IP da estação de trabalho. Os nomes completos de domínio são aceitos.

Para nomes de host, os caracteres válidos são alfanuméricos, incluindo hífen (-). O comprimento máximo é de 51 caracteres.

Especifique `NULL` quando:

- Definindo um agente estendido para PeopleSoft, SAP ou Oracle.

- **host** for configurado para `$MASTER`

Se estiver definindo uma estação de trabalho de mecanismo remoto, especifique o nome do host do sistema onde o mecanismo remoto está instalado.

tcpaddr *port*
Especifica o número da porta TCP/IP **netman** que o Tivoli Workload Scheduler utiliza para comunicação entre as estações de trabalho.

Para estações de trabalho do Workload Broker
Especifique o valor da propriedade **TWS.Agent.Port** do arquivo `TWSAgentConfig.properties`.

Para estações de trabalho de mecanismo remoto usando o protocolo HTTP para se comunicar com o mecanismo remoto
Especifique o número da porta HTTP do mecanismo remoto.

Para outros tipos de estações de trabalho
Especifique o valor designado no arquivo `localopts` para a variável *nm port*.

O valor padrão para esse campo é `31111`. Especifique um valor no intervalo de `1` a `65535`.

secureaddr
Define a porta utilizada para atender conexões SSL recebidas. Esse valor é lido quando o atributo **securitylevel** for configurado.

Para estações de trabalho do Workload Broker
Ignore esse atributo.

Para estações de trabalho de mecanismo remoto usando o protocolo HTTPS para se comunicar com o mecanismo remoto
Especifique o número da porta HTTPS do mecanismo remoto.

Para outros tipos de estações de trabalho
Especifique o valor designado no arquivo `localopts` para a variável *nm ssl port*. O valor deve ser diferente do valor designado à variável *nm port* no arquivo `localopts`.

Se **securitylevel** for especificado, mas esse atributo estiver ausente, o valor padrão para esse campo é `31113`. Especifique um valor no intervalo de `1` a `65535`.Consulte o *IBM Tivoli Workload Scheduler Administration and Installation Guide* para obter informações sobre autenticação SSL e opções locais configuradas no arquivo de configuração *TWS_home*`/localopts`.

timezone|tz *tzname*
Especifica o fuso horário da estação de trabalho. Para assegurar a precisão dos horários de planejamento, esse fuso horário deve ser o mesmo contido no sistema operacional do computador.

Quando usado em definições da estação de trabalho do mecanismo remoto, representa o fuso horário no mecanismo remoto do Tivoli Workload Scheduler.

Consulte Capítulo 16, “Gerenciando fusos horários”, na página 597 para obter os nomes dos fusos horários válidos.

domain *domainname*
Especifica o nome do domínio do Tivoli Workload Scheduler ao qual a estação de trabalho pertence. O valor padrão para estação de trabalho tolerante a falhas é `MASTERDM`.

O Tivoli Workload Scheduler ignora a configuração do domínio quando definido para agentes estendidos, exceto quando o atributo**host** for configurado para `$MASTER`.

Essa configuração é obrigatória para estações de trabalho de agente padrão e do dynamic workload broker.

host *host-workstation*
Esse atributo é obrigatório para agentes estendidos e estações de trabalho de mecanismo remoto e especifica:

Para estações de trabalho de mecanismo remoto, agentes, conjuntos e conjuntos dinâmicos:
A estação de trabalho de broker hosting a estação de trabalho. Esse campo não pode ser atualizado após a criação da estação de trabalho de mecanismo remoto.

Para agentes estendidos
A estação de trabalho com a qual o agente estendido se comunica e onde seu método de acesso está instalado. A estação de trabalho hosting não pode ser outro agente estendido.

Se a estação de trabalho hosting for um gerenciador de domínio para o qual definiu um backup, é possível especificar um dos valores a seguir para assegurar que o agente estendido não seja isolado se a estação de trabalho hosting ficar indisponível.

$MASTER
Para indicar que a estação de trabalho host para o agente estendido é ogerenciador de domínio principal.

$MANAGER
Para indicar que a estação de trabalho host para o agente estendido é o gerenciador de domínio. Nesse caso, você deve especificar o domínio onde o agente está localizado.

Nesse caso, certifique-se de que os métodos do agente estendido estejam instalados também na estação de trabalho de backup. É possível ativar e desativar a resolução automática da chave `$MASTER` usando a opção *mm resolve master* no arquivo `localopts`.

Para obter informações adicionais sobre as opções disponíveis no arquivo `localopts`, consulte o *Guia de Administração do IBM Tivoli Workload Scheduler*.

access *method*
Especifica um método de acesso para agentes estendidos e agentes de rede. Corresponde ao nome de um arquivo que está localizado no diretório *TWS_home*`/methods` na estação de trabalho hosting.

Especifique `NULL` ao definir um agente estendido para PeopleSoft, SAP ou Oracle.

agentID *agentID*
O identificador exclusivo do agent.

tipo Especifica o tipo da estação de trabalho. Se planeja alterar os tipos de estação de trabalho, considere as regras a seguir:

- É possível alterar o agente tolerante a falhas, o agente padrão, o agente estendido, o gerenciador de domínio e as estações de trabalho do broker

de carga de trabalho dinâmico para qualquer tipo de estação de trabalho, com exceção do agente dinâmico, conjunto, conjunto dinâmico e mecanismo remoto.

- Não é possível alterar o tipo de agente dinâmico, conjunto, conjunto dinâmico e mecanismo remoto.

Insira um dos valores a seguir:

fta Se você definir um *agente tolerante a falhas*, ou seja uma estação de trabalho do agente que ativa tarefas e resolva dependências locais sem um gerenciador de domínio. Esse é o valor padrão para esse atributo.

Você deve especificar `fta` se quiser designar à estação de trabalho a função de gerenciador do domínio principal de backup ou de gerenciador de domínio principal de backup.

Especifique `fta` para sistemas IBM i executando como agentes tolerantes a falhas limitados.

s-agent
Se definir um *agente padrão*, ou seja uma estação de trabalho do agente que ative tarefas somente sob a direção de seu gerenciador de domínio.

agente x
Se definir um *agente estendido*, ou seja uma estação de trabalho do agente que ative tarefas somente sob a direção de sua estação de trabalho hosting. Os agentes estendidos podem ser utilizados para fazer interface do Tivoli Workload Scheduler com sistemas e aplicativos não-Tivoli.

Para obter informações adicionais, consulte *IBM Tivoli Workload Scheduler for Applications*.

manager
Se você definir um *gerenciador de domínio*, ou seja uma estação de trabalho que gerencia um domínio. Ao definir esse tipo de estação de trabalho, especifique:

Servidor
NULL

Domínio
Onome do domínio que a estação de trabalho gerencia, se diferente do domínio`MASTERDM`.

Você especifica que uma estação de trabalho é um gerenciador também no campo *manager* da "Definição de domínio" na página 154. O Tivoli Workload Scheduler verifica automaticamente se os valores especificados nesses campos estão consistentes.

intermediário
Se você definir uma estação de trabalho do *dynamic workload broker*, essa será uma estação de trabalho que executa os tipos de tarefas existentes e otipos de tarefas com opções avançadas. É o servidor do broker instalado com o gerenciador de domínio principal e ogerenciador de domínio dinâmico. Hospeda as seguintes estações de trabalho:

- Agente Estendido
- Mecanismo remoto

- Conjunto
- Conjunto dinâmico
- Agente. Essa definição inclui os seguintes agentes:
  - agent
  - Agente do Tivoli Workload Scheduler for z/OS
  - Agent for z/OS

  Para obter informações adicionais sobre oagent e oAgente do Tivoli Workload Scheduler for z/OS, consulte *Planejando Carga de Trabalho Dinamicamente*. Para obter informações adicionais sobre oAgent for z/OS, consulte *Planejando com o Agent for z/OS*.

agent Se você definir um *agente dinâmico*, que é uma estação de trabalho que gerencia uma ampla variedade de tipos de tarefa, por exemplo, banco de dados específico ou tarefas de transferência de arquivos, além de tipos de tarefas tradicionais. Ela é hospedada pela workload broker workstation. A definição da estação de trabalho é criada e registrada automaticamente ao instalar o componente do agente dinâmico. Em sua definição, é possível editar somente os atributos a seguir:

- Descrição
- Vartable

Nota: Se você tiver a opção global `enAddWorkstation` configurada como `"yes"`, a definição de estação de trabalho do agente dinâmico é automaticamente incluído no Plano após o processo de instalação criar a estação de trabalho do agente dinâmico no banco de dados.

rem-eng
Se você definir uma *estação de trabalho do mecanismo remoto*, que é uma estação de trabalho usada para se comunicar com um mecanismo remoto ao ligar uma tarefa definida localmente, denominada *tarefa sombra*, a uma tarefa específica em execução no mecanismo remoto, denominada *tarefa remota*. Quando as duas tarefas são conectadas, a transição do status da tarefa sombra mapeia a transição do status da tarefa remota. Esse mapeamento é útil também para definir e monitorar dependências de tarefas locais em tarefas em execução no mecanismo remoto; essas dependências são chamadas *dependências cruzadas*.

Para obter informações adicionais sobre tarefas sombra e dependências cruzadas, consulte Capítulo 19, “Definindo e Gerenciando Dependências Cruzadas”, na página 625.

Ao definir esse tipo de estação de trabalho, especifique os seguintes campos:

os O sistema operacional do mecanismo remoto.

host O nome da estação de trabalho do broker hosting.

nó O nome do host ou o endereço IP do mecanismo remoto.

Ao especificar o número da porta a ser usado para comunicação com o mecanismo remoto, use **secureaddr** se o protocolo usado for HTTPS, **tcpaddr** se o protocolo usado for HTTP. Recomenda-se especificar no campo **timezone** o fuso horário configurado no mecanismo remoto.

conjunto
Se você definir um *conjunto*, ou seja um conjunto de agentes dinâmicos com características de hardware ou de software semelhantes para o qual enviar tarefas. Essa estação de trabalho é hospedada pela workload broker workstation. Em sua definição, é possível editar somente os atributos a seguir:

- Descrição
- Vartable
- Membros

d-pool Se você definir um *conjunto dinâmico*, ou seja um conjunto de agentes dinâmicos definidos dinamicamente com base nos requisitos listados no arquivo JSDL especificados no atributo **resources**. Essa estação de trabalho é hospedada pela workload broker workstation. Em sua definição, é possível editar somente os atributos a seguir:

- Descrição
- Vartable
- Requisitos

ignorar
Especifica que a definição da estação de trabalho não deve ser incluída no plano de produção. Se você especificar esta configuração, os fluxos de tarefas planejados para execução nesta estação de trabalho não serão incluídos no plano de produção.

autolink
Especifica se o link deve ser aberto entre as estações de trabalho na inicialização. Dependendo do tipo da estação de trabalho, ao configurar seu valor para `on`:

Em um agente tolerante a falhas ou em um agente padrão
Significa que o gerenciador de domínio abre o link para o agente quando o gerenciador de domínio for iniciado.

Em um gerenciador de domínio
Significa que seus agentes abrem links para o gerenciador de domínio quando eles forem iniciados.

Essa configuração é especialmente útil quando um novo plano de produção for criado no gerenciador de domínio principal: Como parte da geração do plano de produção, todas as estações de trabalho são paradas e, em seguida, iniciadas. Para cada agente com **autolink** ativado, o gerenciador de domínio envia automaticamente uma cópia do novo plano de produção e, em seguida, inicia o agente. Se o**autolink** estiver ativado também para o gerenciador de domínio, o agente abre um link de volta para o gerenciador de domínio.

Se o valor do **autolink** for `off` para o agente, é possível abrir o link de seu gerenciador de domínio executando o comando **conman link** no gerenciador de domínio do agente ou no gerenciador de domínio principal.

behindfirewall
Se configurado para `on`, significa que há um firewall entre a estação de trabalho e o gerenciador de domínio principal. Nesse caso, somente uma conexão direta entre a estação de trabalho e seu gerenciador de domínio é permitia e os comandos **start** *workstation*, **stop** *workstation* e **showjobs** são enviados seguindo a hierarquia do domínio, em vez de fazer com que o

gerenciador de domínio principal ou o gerenciador de domínio abra uma conexão direta com a estação de trabalho.

Configure esse atributo para `off` se estiver definindo uma estação de trabalho com o tipo `broker`.

fullstatus
Especifique esta configuração ao definir uma estação de trabalho do agente tolerante a falhas. Para gerenciadores de domínios, essa palavra-chave é configurada automaticamente para `on`. Especifique um dos seguintes valores:

em Se desejar que a estação de trabalho do agente tolerante a falhas opere no modo*status completo*, significando que a estação de trabalho é atualizada com o status de tarefas e de fluxos de tarefas em execução em todas as outras estações de trabalho em seu domínio e em domínios subordinados, mas não em domínios peer ou pai. A cópia do plano de produção no agente é mantida no mesmo nível de detalhe que a cópia do plano de produção em seu gerenciador de domínio.

off Se desejar que a estação de trabalho do agente tolerante a falhas seja informada sobre o status de tarefas e fluxos de tarefas apenas em outras estações de trabalho que afetam suas próprias tarefas e fluxos de tarefas. Especificar "on" pode melhorar o desempenho reduzindo a atividade de rede.

securitylevel
Especifica o tipo de autenticação SSL para a estação de trabalho. Não especifique esse atributo para uma estação de trabalho com tipo `broker`. Pode ter um dos seguintes valores:

ativado
A estação de trabalho utiliza a autenticação SSL apenas se requerido por sua estação de trabalho do gerenciador de domínio ou por outro agente tolerante a falhas abaixo dele na hierarquia de domínio.

em A estação de trabalho utiliza a autenticação SSL quando se conecta com seu gerenciador de domínio. O gerenciador de domínio utiliza a autenticação SSL quando se conecta a seu gerenciador de domínio pai. O agente tolerante a falhas recusa qualquer conexão recebida de seu gerenciador de domínio se não for uma conexão SSL.

forçar A estação de trabalho utiliza a autenticação SSL para todas as suas conexões e aceita conexões dos seus gerenciadores de domínio pai e subordinados. A estação de trabalho recusa qualquer conexão recebida que não seja uma conexão SSL.

Se esse atributo for omitido, a estação de trabalho não é configurada para conexões SSL e qualquer valor para **secureaddr** é ignorado. Certifique-se, nesse caso, de que a opção local *nm ssl port* esteja configurada para 0 para assegurar que o processo **netman** não tente abrir a porta especificada em **secureaddr**. Consulte o *IBM Tivoli Workload Scheduler Administration and Installation Guide* para obter informações sobre autenticação SSL.

A seguinte tabela descreve o tipo de comunicação utilizada para cada tipo de definição **securitylevel**.

Tabela 26. Tipo de Comunicação, Dependendo do Valor do Nível de Segurança

Valor configurado no Agente Tolerante a Falhas (ou no Gerenciador de Domínio)	Valor configurado em seu Gerenciador de Domínio (ou em seu Gerenciador de Domínio Pai)	Tipo de conexão estabelecida
Não especificado	Não especificado	TCP/IP
Ativado	Não especificado	TCP/IP
Em	Não especificado	Sem conexão
Force	Não especificado	Sem conexão
Não especificado	Em	TCP/IP
Ativado	Em	TCP/IP
Em	Em	SSL
Force	Em	SSL
Não especificado	Ativado	TCP/IP
Ativado	Ativado	TCP/IP
Em	Ativado	SSL
Force	Ativado	SSL
Não especificado	Force	Sem conexão
Ativado	Force	SSL
Em	Force	SSL
Force	Force	SSL

O valor para **securitylevel** não é especificado para estações de trabalho do broker de carga de trabalho dinâmica ou para estações de trabalho com um agente do Tivoli Workload Scheduler V8.2 ou anterior.

server *ServerID*

Use o atributo**server** na definição da estação de trabalho do agente tolerante a falhas para reduzir o tempo necessário para inicializar agentes e para melhorar o tempo das mensagens. Por padrão, comunicações com os agente tolerante a falhass são tratadas por um processo do **mailman** em execução no gerenciador de domínio. O atributo**server** permite iniciar um processo do **mailman** no gerenciador de domínio para tratar de comunicações somente com essa estação de trabalho do agente tolerante a falhas.

Se estiver definindo um agente tolerante a falhas que possa trabalhar como um gerenciador do domínio de backup, o *ServerID* é usado somente quando a estação de trabalho funciona como um agente tolerante a falhas; a configuração é ignorada quando a estação de trabalho funciona como um gerenciador do domínio de backup.

No *ServerID*, o ID é uma única letra ou um número (A-Z e 0-9). Os IDs são exclusivos para cada gerenciador de domínio, de forma que você pode utilizar os mesmos IDs em outros domínios sem conflito. Um *ServerID* específico pode ser dedicado a mais de uma estação de trabalho do agente tolerante a falhas.

Como boas práticas:

- Se mais de 36 IDs de servidor forem necessários em um domínio, considere dividir o domínio em dois ou mais domínios.
- Se o mesmo ID for utilizado para vários agentes, um único servidor será criado para manipular suas comunicações. Defina servidores extras para evitar que um único servidor não manipule mais de oito agentes.

Se um *ServerID* não for especificado, as comunicações com o agente serão manipuladas pelo processo **mailman** principal no gerenciador de domínio.

protocol *http | https*

Especifica o protocolo a usar para se comunicar com:

A estação de trabalho do broker

Se a estação de trabalho for uma estação de trabalho do agente.

O mecanismo remoto
Se a estação de trabalho for uma estação de trabalho de mecanismo remoto.

members *[workstation] [...]*
Use esse valor para uma estação de trabalho de conjunto para especificar os agentes que deseja incluir no conjunto.

requirements *jsdl_definition*
Use esse valor para uma estação de trabalho do conjunto dinâmico para especificar os requisitos, em formato JSDL; os agentes que satisfazem o requisito automaticamente pertencem ao conjunto dinâmico. A sintaxe a seguir é usada:

```
jsdl_definition:
<jsdl:resources>
<jsdl:logicalResource subType="MyResourceType"/>
</jsdl:resources>
```

Para obter informações adicionais sobre a sintaxe JSDL, consulte *Planejando Carga de Trabalho Dinamicamente*.

Nota: É possível incluir definições da estação de trabalho no banco de dados em qualquer momento, mas você deve executar **JnextPlan -for 0000** novamente para poder executar tarefas nas novas estações de trabalho criadas. Toda vez que **JnextPlan** é executado, todas as estações de trabalho são paradas e reiniciadas.

Exemplos

O exemplo a seguir cria um gerenciador de domínio principal denominado`hdq-1` e um agente tolerante a falhas denominado`hdq-2` no domínio principal. Observe que um argumento **domain** é opcional neste exemplo, porque o domínio principal é padronizado como **masterdm**.

```
cpuname hdq-1 description "Headquarters master DM"
     os unix
     tz America/Los_Angeles
     node sultan.ibm.com
     domain masterdm
     for maestro type manager
          autolink on
          fullstatus on
end

cpuname hdq-2
     os wnt
     tz America/Los_Angeles
     node opera.ibm.com
     domain masterdm
     for maestro type fta
          autolink on
end
```

O exemplo a seguir cria um domínio chamado `distr-a` com um gerenciador de domínio chamado `distr-a1` e um agente padrão chamado `distr-a2`:

```
domain distr-a
     manager distr-a1
     parent masterdm
end

cpuname distr-a1 description "District A domain mgr"
     os wnt
     tz America/New_York
     node pewter.ibm.com
     domain distr-a
     for maestro type manager
          autolink on
          fullstatus on
end

cpuname distr-a2
     os wnt
     node quatro.ibm.com
     tz America/New_York
     domain distr-a
     for maestro type s-agent
end
```

O exemplo a seguir cria uma estação de trabalho tolerante a falhas com autenticação SSL. A definição de segurança **securitylevel** especifica que a conexão entre a estação de trabalho e seu gerenciador de domínio pode ser somente do tipo SSL. A porta 32222 é reservada para atender conexões SSL de entrada.

```
cpuname ENNETI3
     os WNT
     node apollo
     tcpaddr 30112
     secureaddr 32222
     for maestro
          autolink off
          fullstatus on
          securitylevel on
end
```

O exemplo a seguir cria uma estação de trabalho de intermediário. Esta estação de trabalho gerencia o ciclo de vida do tipo de workload broker do Tivoli Workload Scheduler no Tivoli dynamic workload broker.

```
cpuname ITDWBAGENT
       vartable TABLE1
       os OTHER
      node itdwbtst11.ibm.com TCPADDR 41114
      timezone Europe/Rome
      domain MASTERDM
      for MAESTRO
          type BROKER
          autolink OFF
          behindfirewall OFF
          fullstatus OFF
end
```

O exemplo a seguir cria uma estação de trabalho de mecanismo remoto a ser usada para gerenciar dependências cruzadas e se comunicar com um mecanismo remoto instalado em um sistema com nome do host `London-hdq` usando a porta HTTPS padrão 31116. A estação de trabalho do mecanismo remoto é hospedada pela estação de trabalho do broker ITDWBAGENT

```
cpuname REW_London
     description "Remote engine workstation to communicate with London-hdq"
     os WNT
     node London-hdq secureaddr 31116
     timezone Europe/London
     for maestro host ITDWBAGENT
        type rem-eng
        protocol HTTPS
end
```

O exemplo a seguir mostra como criar um conjunto dinâmico de agentes. Todos os agentes do conjunto dinâmico devem ter os sistemas operacionais HP-UX ou Linux instalados:

```
CPUNAME DPOOLUNIX
  DESCRIPTION "Sample Dynamic Pool Workstation"
  VARTABLE table1
  OS OTHER
  TIMEZONE Europe/Rome
  FOR MAESTRO HOST MAS86MAS_DWB
    TYPE D-POOL
  REQUIREMENTS
    <?xml version="1.0" encoding="UTF-8"?>
<jsdl:resourceRequirements
      xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl">
  <jsdl:resources>
    <jsdl:candidateOperatingSystems>
      <jsdl:operatingSystem type="HPUX"/>
      <jsdl:operatingSystem type="LINUX"/>
    </jsdl:candidateOperatingSystems>
  </jsdl:resources>
</jsdl:resourceRequirements>
END
```

O exemplo a seguir mostra como criar um conjunto dinâmico de agentes. Todos os agentes do conjunto dinâmico devem ter o sistema operacional Windows 2000 instalado:

```
CPUNAME DPOOLWIN
  DESCRIPTION "Sample Dynamic Pool Workstation"
  OS WNT
  TIMEZONE Europe/Rome
  FOR MAESTRO HOST MAS86MAS_DWB
    TYPE D-POOL
  REQUIREMENTS
    <?xml version="1.0" encoding="UTF-8"?>
<jsdl:resourceRequirements
      xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl">
  <jsdl:resources>
```

```
    <jsdl:candidateOperatingSystems>
      <jsdl:operatingSystem type="Windows 2000"/>
    </jsdl:candidateOperatingSystems>
  </jsdl:resources>
</jsdl:resourceRequirements>
END
```

O exemplo a seguir mostra como criar um conjunto de agentes denominado POOLUNIX e que contém dois agentes: NC121105 e NC117248:

```
CPUNAME POOLUNIX
  DESCRIPTION "Sample Pool Workstation"
  OS OTHER
  TIMEZONE Europe/Rome
  FOR MAESTRO HOST MAS86MAS_DWB
    TYPE POOL
  MEMBERS
    NC121105
    NC117248
END
```

Consulte também

Para obter mais informações sobre como executar a mesma tarefa a partir do Dynamic Workload Console, consulte:

o Guia do Usuário do Console do Dynamic Workload, seção Criando Cargas de Trabalho Distribuídas.

Definição de classe de estação de trabalho

Uma classe de estação de trabalho é um grupo de estações de trabalho para o qual tarefas comuns e fluxos de tarefa podem ser gravados. É possível incluir várias definições de classes de estação de trabalho no mesmo arquivo de texto, junto com definições de estação de trabalho e definições de domínio.

Ao definir classes de estação de trabalho, assegure-se de que as estações de trabalho na classe suportem os tipos de tarefas que planeja executar nelas. As seguintes regras se aplicam:

- Os tipos de tarefas com opções avançadas executam apenas em agentes dinâmicos, conjuntos e conjuntos dinâmicos.
- Tarefas sombra são executadas apenas em mecanismos remotos.

Cada definição de classe de estação de trabalho tem o seguinte formato e argumentos:

Sintaxe

cpuclass *workstationclass*
  [**description** *"description"*]
  [**ignore**]
  members [*workstation* | **@**] [...]
  end

[**cpuname** ...]

[**cpuclass** ...]

[**domain** ...]

Argumentos

cpuclass *workstationclass*
Especifica o nome da classe de estação de trabalho. O nome deve iniciar

com uma letra e pode conter caracteres alfanuméricos, traços e sublinhados. Ele pode conter até 16 caracteres.

Nota: Não é possível utilizar nomes iguais para estações de trabalho, classes de estação de trabalho e domínios.

description *"description"*
Fornece uma descrição da classe de estação de trabalho. A descrição pode conter até 120 caracteres alfanuméricos. O texto deve ser colocado entre aspas duplas.

ignorar
Especifica que o Tivoli Workload Scheduler deve ignorar todas as estações de trabalho incluídas nesta classe de estação de trabalho ao gerar o plano de produção.

members *workstation*
Especifica uma lista de nomes de estações de trabalho, separadas por espaços, que são membros da classe. O caractere curinga @ significa que a classe de estação de trabalho inclui todas as estações de trabalho.

Exemplos

O exemplo a seguir define uma classe de estação de trabalho chamada `backup`:

```
cpuclass backup
     members
          main
          site1
          site2
end
```

O exemplo a seguir define uma classe de estação de trabalho chamada `allcpus` que contém todas as estações de trabalho:

```
cpuclass allcpus
     members
          @
end
```

Consulte também

Para obter mais informações sobre como executar a mesma tarefa a partir do Dynamic Workload Console, consulte:

a seção Guia do Usuário do Console do Dynamic Workload, "Projetando sua Carga de Trabalho".

Definição de domínio

Um domínio é um grupo de estações de trabalho que consiste em um ou mais agentes e em um gerenciador de domínio. O gerenciador de domínio age como um hub de gerenciamento para os agentes no domínio. É possível incluir várias definições de domínio no mesmo arquivo de texto, junto com definições de estação de trabalho e definições de classes de estação de trabalho. Cada definição de domínio tem o seguinte formato e argumentos:

Sintaxe

domain *domainname*[**description** *"description"*]
    *** manager** *workstation*
     [**parent** *domainname* | **ismaster**]
end

[**cpuname** ...]

[**cpuclass** ...]

[**domain** ...]

Argumentos

domain *domainname*
O nome do domínio. Deve iniciar com uma letra e pode conter caracteres alfanuméricos, traços e sublinhados. Ele pode conter até 16 caracteres. Não é possível utilizar nomes iguais para estações de trabalho, classes de estação de trabalho e domínios.

description "*description***"**
Fornece uma descrição do domínio. O texto deve ser colocado entre aspas duplas.

*** manager** *workstation*
Esse é um campo comentado utilizado somente para mostrar, ao exibir a definição do domínio, o nome da estação de trabalho que tem a função do gerenciador de domínio para esse domínio. Certifique-se de que esse campo permaneça como comentário. Ele é mantido para compatibilidade com versões anteriores. Com o Tivoli Workload Scheduler versão 8.3, as informações sobre se uma estação de trabalho é um gerenciador de domínio são configuradas no campo **type** no "Definição de estação de trabalho" na página 139.

parent *domainname*
O nome do domínio pai ao qual o gerenciador de domínio está ligado. O padrão é o domínio principal, que não requer uma definição de domínio. O domínio principal é definido pelas opções globais **master** e **master domain**.

ismaster
Se especificado, indica que o domínio é o domínio principal da rede do Tivoli Workload Scheduler. Se configurado, não pode ser removido posteriormente.

Exemplos

O exemplo a seguir define um domínio chamado `east`, com o domínio principal como seu pai e dois domínios subordinados chamados `northeast` e `southeast`:

```
domain east
     description "O domínio Oriental"
     * manager cyclops
end
domain northeast
     description "O domínio no Nordeste"
     * manager boxcar
     parent east
end
domain southeast
     description "O domínio no Sudeste"
     * manager sedan
     parent east
end
```

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

Guia do Usuário do Console do Dynamic Workload, a seção sobre Criar um domínio.

Definição de tarefa

Um job é um arquivo, programa ou comando executável que é planejado e ativado pelo Tivoli Workload Scheduler. Você pode gravar definições de tarefas nos arquivos de edição e, em seguida, incluí-los no banco de dados do Tivoli Workload Scheduler com o programa criador. É possível incluir várias definições de job em um único arquivo de edição.

Cada definição de tarefa tem o seguinte formato e argumentos:

Sintaxe

$jobs
[*workstation#*]*jobname*
{**scriptname** *filename* **streamlogon** *username* |
docommand "*command*" **streamlogon** *username* |
task *job_definition* [**streamlogon** *username*]}
[**description** "*description*"]
[**tasktype** *tasktype*]
[**interactive**]
[**rccondsucc** "*Success Condition*"]
[**recovery**
{**stop** | **continue** | **rerun**}
[**after** [*workstation#*]*jobname*]
[**abendprompt** "*text*"]]

Uma tarefa em si não possui configurações para dependências, elas devem ser adicionadas a uma tarefa quando incluída em uma definição de fluxo de tarefas.

É possível incluir ou modificar definições de tarefas a partir de definições de fluxos de tarefas. Modificações em definições de tarefas feitas nas definições dos fluxos de tarefas são refletidas nas definições de tarefas armazenadas no banco de dados. Isso significa que se você modificar a definição de tarefa de `job1` na definição de fluxo de tarefas `js1` e `job1` também for utilizado no fluxo de tarefa `js2`, a definição de `job1` na definição `js2` também é modificada conforme necessário.

Nota: Palavras-chave incorretamente digitadas utilizadas em definições de tarefa levam a definições de tarefa truncada armazenadas no banco de dados. De fato, a palavra-chave incorreta é considerada estranha para a definição de tarefa e, portanto, é interpretada como o nome de tarefa de uma definição de tarefa adicional. Geralmente, essa interpretação incorreta também causa um erro de sintaxe ou um erro de definição de tarefa inexistente para a definição de tarefa adicional.

É necessário atenção especial no caso em que um alias foi designado a uma tarefa. É possível decidir usar um nome diferente para referir-se a uma instância da tarefa específica em um fluxo de tarefas, mas o nome do alias não deve entrar em conflito com o nome da tarefa em outra tarefa no mesmo fluxo de tarefas. Se uma definição de tarefa for renomeada, as tarefas que possuem o mesmo nome que a definição de tarefa modificarão o nome, de acordo com o nome da definição de tarefa. A seguir estão alguns exemplos para entender o comportamento de tarefas quando o nome da definição de tarefa é modificado:

Tabela 27. Exemplos: Renomeando a Definição de Tarefa

Nomes da definição de tarefa original no fluxo de tarefas	Renomear definição de tarefa	Resultado
SCHEDULE WKS#JS : FTA1#**A** FTA1#**B como C** END	Renomear tarefa **A** para **D**	SCHEDULE WKS#JS : FTA1#**D** FTA1#**B como C** END
	Renomear tarefa **B** para **D**	SCHEDULE WKS#JS : FTA1#**A** FTA1#**D como C** END
	Renomear tarefa **A** para **C**	Ocorre um erro ao renomear a tarefa **A** para **C** porque a tarefa C já existe como o alias para a tarefa **B**.

Consulte a seção "Definição de fluxo de tarefas" na página 216 para obter informações sobre como gravar definições de fluxos de tarefas.

Argumentos

*workstation***#**

Especifica o nome da estação de trabalho ou classe de trabalho na qual o job é executado. O padrão é a estação de trabalho especificada para *defaultws* ao iniciar a sessão do **composer**.

Para obter informações adicionais sobre como iniciar uma sessão do composer, consulte "Executando o Programa composer" na página 279. O sinal de sustenido (**#**) é um delimitador necessário. Se você especificar uma classe de estação de trabalho, ela deverá corresponder à classe de estação de trabalho de qualquer fluxo de tarefa no qual a tarefa foi incluída.

Se estiver definindo uma tarefa que gerencia uma tarefa de broker da carga de trabalho, especifique o nome da estação de trabalho onde a estação de trabalho do broker da carga de trabalho está instalada. Utilizando a estação de trabalho do broker da carga de trabalho, o Tivoli Workload Scheduler pode submeter a tarefa no ambiente Tivoli dynamic workload broker usando o envio de tarefa dinâmico.

jobname

Especifica o nome do job. O nome deve iniciar com uma letra e pode conter caracteres alfanuméricos, traços e sublinhados. Ele pode conter até 40 caracteres.

scriptname *filename*

Especifique o nome do arquivo no qual o job é executado. Utilize **scriptname** para jobs do UNIX e do Windows. Para um arquivo executável, digite o nome do arquivo e todas as opções e argumentos. O *filename* mais a *Success Condition* (da palavra-chave **rccondsucc**) não devem exceder 4095 caracteres. Os parâmetros do Tivoli Workload Scheduler também podem ser utilizados.

Consulte "Utilizando Variáveis e Parâmetros nas Definições de Tarefa" na página 191 para obter informações adicionais.

Para jobs do Windows, inclua as extensões do arquivo. Nomes UNC (Universal Naming Convention) são permitidos. Não especifique arquivos em unidades mapeadas.

Se estiver definindo uma tarefa que gerencia uma tarefa do broker de carga de trabalho, especifique o nome do host da tarefa do broker da carga de trabalho. Além disso, é possível especificar variáveis e o tipo de afinidade

que existem entre a tarefa Tivoli Workload Scheduler e a tarefa do broker de carga de trabalho usando a sintaxe descrita na lista a seguir. Para identificar uma tarefa relacionada utilizando o:

Nome da Tarefa do Tivoli Workload Scheduler
jobName [-var var1Name=var1Value,...,varNName=varNValue] [-twsAffinity jobname=twsJobName]

ID da tarefa do Tivoli dynamic workload broker
jobName [-var var1Name=var1Value,...,varNName=varNValue] [-affinity jobid=jobid]

Alias de tarefa do Tivoli dynamic workload broker
jobName [-var var1Name=var1Value,...,varNName=varNValue] [-affinity alias=alias]

Consulte o *IBM Tivoli Workload Scheduler: Scheduling Workload Dynamically* para obter informações detalhadas.

Se o caminho do arquivo ou o nome do arquivo do argumento scriptname contiver espaços, a cadeia inteira deve ser inserida entre "\" e \" ", como mostrado abaixo:

```
scriptname "\"C:\Program
Files\tws\myscript.cmd\""
```

Se caracteres especiais forem incluídos, diferentes de barras (/) e barras invertidas (\), a cadeia inteira deverá ser colocada entre aspas (").

A tarefa `falha` se o script especificado na opção **scriptname** não for localizado ou não tiver permissão de execução. Ela `é encerrada de forma anormal` se o script que não for localizado ou não tiver permissão de execução incluir parâmetros.

docommand *command*
Especifica um comando executado pelo job. Insira um comando válido e quaisquer opções e argumentos entre aspas duplas ("). O *command* mais a *Success Condition* (da palavra-chave **rccondsucc**) não devem exceder 4095 caracteres. Os parâmetros do Tivoli Workload Scheduler também podem ser digitados.

A tarefa `é encerrada de forma anormal` se o arquivo especificado com a opção **docommand** não for localizado ou não tiver permissão de execução.

Consulte "Utilizando Variáveis e Parâmetros nas Definições de Tarefa" na página 191 para obter informações adicionais.

tarefa *job_definition*
Especifica a sintaxe XML para tipos de tarefas com opções avançadas e tarefas sombra. Para definir tipos de tarefas existentes, use a palavra-chave **docommand**. Essa palavra-chave se aplica somente a estações de trabalho dos seguintes tipos:

- agent
- conjunto
- d-pool
- rem-eng

A sintaxe da tarefa depende do tipo de tarefa definido.

Para obter uma lista completa de tipos de tarefas suportados, consulte "Criando o tipos de tarefas com opções avançadas" na página 474.

streamlogon*username*

O nome do usuário sob o qual o job é executado. Esse atributo é obrigatório quando **scriptname** ou **docommand** são especificados. O nome pode conter até 47 caracteres. Se o nome contiver caracteres especiais, ele deverá ser colocado entre aspas duplas ("). Especifique um usuário que pode efetuar logon na estação de trabalho na qual o job é executado. Os parâmetros do Tivoli Workload Scheduler também podem ser digitados.

Consulte “Utilizando Variáveis e Parâmetros nas Definições de Tarefa” na página 191 para obter informações adicionais.

Para jobs do Windows, o usuário também deve ter uma definição de usuário.

Consulte “Definição do Usuário” na página 192 para obter requisitos do usuário.

Se estiver definindo uma tarefa que gerencia uma tarefa do dynamic workload broker, especifique o nome do usuário usado para instalar o Tivoli dynamic workload broker.

A tarefa `falha` se o usuário especificado na opção **streamologon** não existir.

description “*description***”**
Fornece uma descrição do job. O texto deve ser colocado entre aspas duplas.O número máximo de caracteres permitidos é 120.

tasktype *tasktype*

Especifica o tipo de tarefa. Pode ter um dos seguintes valores:

UNIX Para tarefas executadas nas plataformas UNIX.

WINDOWS
Para tarefas executadas nas plataformas Windows.

OUTRO
Para tarefas executadas em agentes estendidos. Consulte o *IBM Tivoli Workload Scheduler for Applications: User's Guide* para obter informações sobre tipos de tarefas customizadas para aplicativos de terceiros suportados.

BROKER
Para tarefas que gerenciam o ciclo de vida de uma tarefa do Tivoli dynamic workload broker. Consulte *IBM Tivoli Workload Scheduler: Scheduling Workload Dynamically* para obter informações sobre como usar o Tivoli dynamic workload broker.

interactive

Se estiver definindo uma tarefa que gerencia uma tarefa do Tivoli dynamic workload broker, ignore este argumento. Especifica que a tarefa é executada interativamente no desktop do usuário. Este recurso está disponível somente nos ambientes Windows.

rccondsucc "*Success Condition***"**
Uma expressão que determina o código de retorno (RC) requerido para considerar um job bem-sucedido. A condição de êxito pode ter no máximo 256 caracteres. Essa expressão pode ser uma das seguintes:

COMPLETE_IF_BIND_FAILS
Essa configuração se aplica somente a tarefas sombra. Quando especificada, o status da tarefa sombra é configurado automaticamente para SUCC se a ligação com a tarefa remota falhar.

Expressão de comparação

Especifica os códigos de retorno do job. A sintaxe é:

```
(RC operator operand)
```

RC A palavra-chave RC.

operator

Operador de comparação. Ele pode ter os seguintes valores:

Tabela 28. Operadores de Comparação

Exemplo	Operator	Descrição
RC<a	<	Menor que
RC<=a	<=	Menor ou igual a que
RC>a	>	Maior que
RC>=a	>=	Maior ou igual a que
RC=a	=	Igual a
RC!=a	!=	Diferente de
RC<>a	<>	Diferente de

operand

Um inteiro entre -2147483647 e 2147483647.

Por exemplo, é possível definir um job bem sucedido como um job que finaliza com um código de retorno menor ou igual a 3, como segue:

```
rccondsucc "(RC <= 3)"
```

Expressão booleana

Especifica uma combinação lógica de expressões de comparação. A sintaxe é:

```
comparison_expression operator comparison_expression
```

comparison_expression

A expressão é avaliada da esquerda para a direita. Você pode utilizar parênteses para atribuir um prioridade à avaliação da expressão.

operator

Operador lógico. Ele pode ter os seguintes valores:

Tabela 29. Operadores lógicos

Exemplo	Operator	Resultado
expr_a e expr_b	And	TRUE se expr_a e expr_b forem TRUE.
expr_a ou expr_b	Ou	TRUE se expr_a ou expr_b for TRUE.
Not expr_a	Not	TRUE se expr_a não for TRUE.

Por exemplo, é possível definir um job bem sucedido como um job que finaliza com um código de retorno menor ou igual a 3 ou com um código de retorno diferente de 5 e menor que 10, como segue:

```
rccondsucc "(RC≤3) OR ((RC≠5) AND (RC<10))"
```

recovery

Recupera opções para o job. O padrão é **stop** sem job de recuperação e sem prompt de recuperação. Digite uma das opções de recuperação, **stop**, **continue** ou **rerun**. Elas podem vir seguidas de um job de recuperação, um prompt de recuperação ou de ambos.

parar Se a tarefa for finalizada anormalmente, não continue com a próxima tarefa.

continuar
Se a tarefa terminar com erro, não continue com a próxima tarefa. A tarefa não é listada como encerrada de forma anormal nas propriedades do fluxo de tarefas. Se nenhum outro problema ocorrer, o fluxo de tarefas é concluído com sucesso.

executar novamente
Se a tarefa for finalizada anormalmente, reexecute a tarefa.

after [*workstation***#]***jobname*

Especifica o nome de uma tarefa de recuperação para execução se a tarefa pai finalizar anormalmente. Os jobs de recuperação são executados somente uma vez para cada instância terminada de forma anormal do job pai.

É possível especificar a estação de trabalho de tarefa de recuperação se ela for diferente da estação de trabalho da tarefa-pai. O padrão é a estação de trabalho da tarefa-pai. Nem todos os jobs são elegíveis para executarem os jobs de recuperação em uma estação de trabalho diferente. Siga estas instruções:

- Se uma das estações de trabalho for um agente estendido, ela deverá ser alojada por um gerenciador de domínio ou por um agente de tolerância a falhas com um valor de **on** para **fullstatus**.
- A estação de trabalho da tarefa de recuperação pode estar no mesmo domínio que a estação de trabalho da tarefa-pai ou em um domínio mais alto.
- Se a estação de trabalho do job de recuperação for um agente tolerante a falhas, ela deverá ter um valor de **on** para **fullstatus**.

abendprompt "*text***"**
Especifica o texto de um prompt de recuperação, colocado entre aspas duplas, a ser exibido se a tarefa finalizar anormalmente. O texto pode conter até 64 caracteres. Se o texto começar com dois pontos (:), o prompt será exibido, mas não será necessária nenhuma resposta para continuar o processamento. Se o texto começar com um ponto de exclamação (!), o prompt será exibido, mas não será gravado no arquivo de log. Os parâmetros do Tivoli Workload Scheduler também podem ser utilizados.

Consulte "Utilizando Variáveis e Parâmetros nas Definições de Tarefa" na página 191 para obter informações adicionais.

A Tabela 30 resume todas as combinações possíveis de ações e opções de recuperação.

A tabela baseia-se nos seguintes critérios de um fluxo de job chamado `sked1`:

- O fluxo de tarefas `sked1` possui duas tarefas, `job1` e `job2`.
- Se for selecionada para `job1`, a tarefa de recuperação será `jobr`.
- `job2` é dependente de `job1` e não é iniciada até `job1` ter sido concluída.

Tabela 30. Opções e Ações de Recuperação

	Parar	Continuar	Executar Novamente
Prompt de recuperação: Não **Job de recuperação**: Não	A intervenção é requerida.	Execute o `job2`.	Execute novamente o `job1`. Se `job1` finalizar anormalmente, emita um prompt. Se a resposta for *sim*, repita a instrução anterior. Se o `job1` for bem-sucedido, execute o `job2`.

Tabela 30. Opções e Ações de Recuperação (continuação)

	Parar	**Continuar**	**Executar Novamente**
Prompt de recuperação: Sim **Job de recuperação**: Não	Emita o prompt de recuperação. A intervenção é requerida.	Emita o prompt de recuperação. Se a resposta for *sim*, execute o `job2`.	Emita o prompt de recuperação. Se a resposta for *sim*, execute novamente o `job1`. Se `job1` finalizar anormalmente, repita acima. Se o `job1` for bem-sucedido, execute o `job2`.
Prompt de recuperação: Não **Job de recuperação**: Sim	Execute o `jobr`. Se finalizar anormalmente, é necessária intervenção. Se for bem-sucedido, execute o `job2`.	Execute o `jobr`. Execute o `job2`.	Execute o `jobr`. Se `jobr` finalizar anormalmente, é necessária intervenção. Se o `jobr` for bem-sucedido, execute novamente o `job1`. Se `job1` finalizar anormalmente, emita um prompt. Se a resposta for *sim*, repita a instrução anterior. Se o `job1` for bem-sucedido, execute o `job2`.
Prompt de recuperação: Sim **Job de recuperação**: Sim	Emita o prompt de recuperação. Se a resposta for *sim*, execute o `jobr`. Se finalizar anormalmente, é necessária intervenção. Se for bem-sucedido, execute o `job2`.	Emita o prompt de recuperação. Se a resposta for *sim*, execute o `jobr`. Execute o `job2`.	Emita o prompt de recuperação. Se a resposta for *sim*, execute o `jobr`. Se `jobr` finalizar anormalmente, é necessária intervenção. Se o `jobr` for bem-sucedido, execute novamente o `job1`. Se `job1` finalizar anormalmente, repita acima. Se o `job1` for bem-sucedido, execute o `job2`.

Notas:

1. "Intervenção é requerida" significa que o `job2` não é liberado de sua dependência do `job1` e, portanto, deve ser liberado pelo operador.
2. A opção de recuperação **continue** substitui o estado de finalização anormal, que pode fazer com que o fluxo de tarefa que contém a tarefa finalizada anormalmente seja marcada como bem-sucedida. Isso evita que o fluxo de tarefas seja transportado para o próximo plano de produção.
3. Se você selecionar a opção **rerun** sem fornecer um prompt de recuperação, Tivoli Workload Scheduler gerará seu próprio prompt.
4. Para fazer referência a um job de recuperação no **conman**, utilize o nome do job original (`job1` no cenário anterior, não `jobr`). Apenas uma tarefa de recuperação é executada para cada finalização anormal.

Exemplos

A seguir, um exemplo de arquivo contendo duas definições de job:

```
$jobs
cpu1#gl1
     scriptname "/usr/acct/scripts/gl1"
     streamlogon acct
     description "general ledger job1"
bkup
     scriptname "/usr/mis/scripts/bkup"
     streamlogon "^mis^"
     recovery continue after recjob1
```

O exemplo a seguir mostra como definir a tarefa Tivoli Workload Scheduler TWSJOB que gerencia a tarefa do workload broker broker_1 que é executada no mesmo agente do workload broker em que TWSJOB2 foi executada:

```
ITDWBAGENT#TWSJOB
SCRIPTNAME "broker_1 -var var1=name,var2=address
           -twsaffinity jobname=TWSJOB2"
STREAMLOGON brkuser
DESCRIPTION "Added by composer."
TASKTYPE BROKER
RECOVERY STOP
```

O exemplo a seguir mostra como definir uma tarefa designada a um conjunto dinâmico de agentes do Windows e executa o script `df`:

```
DPOOLUNIX#JOBDEF7
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
    <jsdl:jobDefinition
          xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
```

```
          xmlns:jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle">
    <jsdl:application name="executable">
    <jsdle:executable interactive="false">
    <jsdle:script>df</jsdle:script>
    </jsdle:executable>
    </jsdl:application>
    </jsdl:jobDefinition>
 DESCRIPTION "Added by composer."
 RECOVERY STOP
```

O exemplo a seguir mostra como definir uma tarefa designada a um conjunto dinâmico de agentes do Windows e executa o script `dir`:

```
DPOOLWIN#JOBDEF6
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
    <jsdl:jobDefinition
          xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
          xmlns:jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle">
    <jsdl:application name="executable">
    <jsdle:executable interactive="false">
    <jsdle:script>dir</jsdle:script>
    </jsdle:executable>
    </jsdl:application>
    </jsdl:jobDefinition>
 DESCRIPTION "Added by composer."
 RECOVERY STOP
```

O exemplo a seguir mostra como definir uma tarefa que é designada ao agent NC115084 e executa o script `dir`:

```
NC115084#JOBDEF3
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
    <jsdl:jobDefinition
          xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
          xmlns:jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle">
    <jsdl:application name="executable">
    <jsdle:executable interactive="false">
    <jsdle:script>dir</jsdle:script>
    </jsdle:executable>
    </jsdl:application>
    </jsdl:jobDefinition>
 DESCRIPTION "Added by composer."
 RECOVERY STOP
```

O exemplo a seguir mostra como definir uma tarefa designada a um conjunto de agentes do UNIX e executa o script definido na tag `script`:

```
POOLUNIX#JOBDEF5
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
    <jsdl:jobDefinition
          xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
          xmlns:jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle">
    <jsdl:application name="executable">
    <jsdle:executable interactive="false">
    <jsdle:script>#!/bin/sh
sleep 60
dir</jsdle:script>
    </jsdle:executable>
    </jsdl:application>
    </jsdl:jobDefinition>
 DESCRIPTION "Added by composer."
 RECOVERY STOP
```

O exemplo a seguir mostra como definir uma tarefa designada a um conjunto de agentes do Windows e executa o script definido na tag `script`:

```
POOLWIN#JOBDEF4
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
    <jsdl:jobDefinition
          xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
          xmlns:jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle">
    <jsdl:application name="executable">
    <jsdle:executable interactive="false">
    <jsdle:script>ping -n 120 localhost</jsdle:script>
    </jsdle:executable>
    </jsdl:application>
    </jsdl:jobDefinition>
 DESCRIPTION "Added by composer."
 RECOVERY STOP
```

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

o Guia do Usuário do Console do Dynamic Workload, seção Criando Definições de Tarefa.

Para obter informações adicionais sobre como criar e editar objetos de planejamento, consulte:

o Guia do Usuário do Console do Dynamic Workload, seção Projetando sua Carga de Trabalho.

Definição de Tarefa - Tarefas Sombra

Tarefas sombra são definidas usando a sintaxe XML. Os atributos-chave para identificar a instância da tarefa remota e os critérios de correspondência dependem do tipo de mecanismo remoto em que a instância de tarefa remota é definida. Campos destacados em negrito são aqueles usados para identificar a instância da tarefa remota.

Como os mecanismos do z/OS suportam somente os critérios e correspondência anteriores mais próximos, o modelo XML para definir uma tarefa sombra do z/OS é o seguinte:

```
$JOBS
WORKSTATION#ZSHADOW_CLOS_PRES
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
    <jsdl:jobDefinition
       xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
       xmlns:zshadow="http://www.ibm.com/xmlns/prod/scheduling/1.0/zshadow">
    <jsdl:application name="zShadowJob">
    <zshadow:ZShadowJob>
    <zshadow:JobStream>JobStream</zshadow:JobStream>
    <zshadow:JobNumber>JobNumber</zshadow:JobNumber>
    <zshadow:matching>
    <zshadow:previous/>
    </zshadow:matching>
    </zshadow:ZShadowJob>
    </jsdl:application>
    </jsdl:jobDefinition>
 DESCRIPTION "Sample Job Definition"
 RECOVERY STOP
```

Nota: Certifique-se de que tenha inserido configurações válidas nos campos **JobStream** e **JobNumber**.

Tarefas sombra distribuídas, no entanto, suportam os quatro critérios de correspondência disponíveis para dependências de sequências externas. A seguir são mostrados modelos XML que é possível usar para definir tarefas sombra distribuídas:

Critérios de correspondência: Mais próximo anterior

Amostra de XML:

```
$JOBS
WORKSTATION#DSHADOW_CLOS_PRES
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
    <jsdl:jobDefinition
       xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
       xmlns:dshadow="http://www.ibm.com/xmlns/prod/scheduling/1.0/dshadow">
    <jsdl:application name="distributedShadowJob">
    <dshadow:DistributedShadowJob>
    <dshadow:JobStream>JobStream</dshadow:JobStream>
    <dshadow:Workstation>Workstation</dshadow:Workstation>
    <dshadow:Job>Job</dshadow:Job>
    <dshadow:matching>
    <dshadow:previous/>
    </dshadow:matching>
```

```
   </dshadow:DistributedShadowJob>
   </jsdl:application>
   </jsdl:jobDefinition>
 DESCRIPTION "Sample Job Definition"
 RECOVERY STOP
```

Critérios de correspondência: Dentro de um intervalo absoluto

Amostra de XML:

```
$JOBS
WORKSTATION#DSHADOW_ABSOLUTE
 TASK
   <?xml version="1.0" encoding="UTF-8"?>
   <jsdl:jobDefinition
         xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
         xmlns:dshadow="http://www.ibm.com/xmlns/prod/scheduling/1.0/dshadow">
   <jsdl:application name="distributedShadowJob">
   <dshadow:DistributedShadowJob>
   <dshadow:JobStream>JobStream</dshadow:JobStream>
   <dshadow:Workstation>Workstation</dshadow:Workstation>
   <dshadow:Job>Job</dshadow:Job>
   <dshadow:matching>
   <dshadow:absolute from="0600 -4" to="1100 +3"/>
   </dshadow:matching>
   </dshadow:DistributedShadowJob>
   </jsdl:application>
   </jsdl:jobDefinition>
 DESCRIPTION "Sample Job Definition"
 RECOVERY STOP
```

Critérios de correspondência: Dentro de um intervalo relativo

```
$JOBS
WORKSTATION#DSHADOW_RELATIVE
 TASK
   <?xml version="1.0" encoding="UTF-8"?>
   <jsdl:jobDefinition
         xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
         xmlns:dshadow="http://www.ibm.com/xmlns/prod/scheduling/1.0/dshadow">
   <jsdl:application name="distributedShadowJob">
   <dshadow:DistributedShadowJob>
   <dshadow:JobStream>JobStream</dshadow:JobStream>
   <dshadow:Workstation>Workstation</dshadow:Workstation>
   <dshadow:Job>Job</dshadow:Job>
   <dshadow:matching>
   <dshadow:relative from="-400" to="+500" />
   </dshadow:matching>
   </dshadow:DistributedShadowJob>
   </jsdl:application>
   </jsdl:jobDefinition>
 DESCRIPTION "Sample Job Definition"
 RECOVERY STOP
```

Critério de correspondência: Mesma data planejada

Amostra de XML:

```
$JOBS
WORKSTATION#DSHADOW_SAMEDAY
 TASK
   <?xml version="1.0" encoding="UTF-8"?>
   <jsdl:jobDefinition
         xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
         xmlns:dshadow="http://www.ibm.com/xmlns/prod/scheduling/1.0/dshadow">
   <jsdl:application name="distributedShadowJob">
   <dshadow:DistributedShadowJob>
   <dshadow:JobStream>JobStream</dshadow:JobStream>
   <dshadow:Workstation>Workstation</dshadow:Workstation>
   <dshadow:Job>Job</dshadow:Job>
   <dshadow:matching>
   <dshadow:sameDay/>
   </dshadow:matching>
   </dshadow:DistributedShadowJob>
   </jsdl:application>
   </jsdl:jobDefinition>
 DESCRIPTION "Sample Job Definition"
 RECOVERY STOP
```

Para obter informações adicionais sobre os critérios de correspondência, consulte “Gerenciando Dependências de Acompanhamentos Externos para Tarefas e Fluxos de Tarefas” na página 61.

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

o Guia do Usuário do Console do Dynamic Workload, seção Criando Definições de Tarefa.

Para obter informações adicionais sobre como criar e editar objetos de planejamento, consulte:

o Guia do Usuário do Console do Dynamic Workload, seção Projetando sua Carga de Trabalho.

Definição de Tarefa - Tarefas de Serviços da Web

Esta seção descreve os atributos necessários e opcionais para tarefas de serviços da Web. As tarefas de serviços da web funcionam apenas com parâmetros de entrada e saída que usam tipos de dados simples. Cada definição de tarefa tem o seguinte formato e argumentos:

Tabela 31. Atributos necessários e opcionais para a definição de uma tarefa de serviços da web

Atributo	Descrição/valor	Exigido
nome do aplicativo	ws	✓
operação	O nome do comando de serviço da Web que você está chamando.	✓
wsdlURL	O nome da URL de serviço da Web.	✓
argumentos	Os parâmetros necessários para o comando de serviço da Web que você está chamando (o número de valores depende do comando).	

credenciais	O nome e a senha do usuário que está executando essa tarefa. Como uma alternativa aos valores reais de codificação permanente, é possível parametrizar de uma das seguintes maneiras:
	• Insira um *nome de usuário* especificado no banco de dados com a definição de usuário (aplicável a todos os sistemas operacionais nesse tipo de tarefa) e insira a instrução:
	`<jsdl:password>${password:username}</jsdl:password>`
	A senha é recuperada da definição de usuário do *nome de usuário* no banco de dados e resolvida em tempo de execução. Consulte “Utilizando Definições do Usuário sobre Tipos de Tarefas com Opções Avançadas” na página 195 para obter mais detalhes.
	Também é possível especificar o usuário de uma estação de trabalho diferente e usar a seguinte sintaxe para a senha:
	`<jsdl:password>${password:workstation#username}</jsdl:password>`
	• Insira um usuário e senha definidos com o comando do utilitário `param` localmente no agente dinâmico que executará a tarefa (se a tarefa for enviada para um conjunto ou para um conjunto dinâmico, a definição deve estar presente em todos os agentes do conjunto). Desde que você tenha definido o nome do usuário com a variável *user* e uma senha, as instruções da credencial correspondente seriam:
	`<jsdl:userName>${agent:user}</jsdl:userName>`
	`<jsdl:password>${agent:password.user}</jsdl:password>`
	As variáveis usuário e senha são resolvidas no agente no tempo de execução. Consulte “Definindo Variáveis e Senhas para Resolução Local em Agentes Dinâmicos” na página 477 para obter mais detalhes.
	Se você usar uma conexão HTTPS, assegure que os certificados de segurança sejam configurados para o Gerenciador de Tarefa na estação de trabalho onde a tarefa deve ser executada.	

A saída do comando é registrada no log da tarefa.

O exemplo a seguir mostra uma tarefa que executa o comando de serviços da Web `getSum`. A definição da tarefa fornece na seção `arguments` os dois valores que devem ser incluídos.

```
$JOBS
AGENT#WEB_SERVICE
TASK
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdlws="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdlws" description="Calls a web
service to perform a sum of two numbers" name="SumNumber">
<jsdl:annotation>annotation</jsdl:annotation>
<jsdl:variables>
<jsdl:stringVariable description="URL of Web Service"
name="wsdlURL">http://np515516.cyber.com:9080/
Sum/services/Sum/wsdl/Sum.wsdl</jsdl:stringVariable>
<jsdl:stringVariable description="Operation to Invoke"
name="Operation">getSum</jsdl:stringVariable>
</jsdl:variables>
<jsdl:application name="ws">
<jsdlws:ws>
<jsdlws:wsToInvoke operation="${Operation}" wsdlURL="${wsdlURL}">
<jsdlws:arguments>
<jsdlws:value>1</jsdlws:value>
<jsdlws:value>2</jsdlws:value>
</jsdlws:arguments>
<jsdlws:credentials>
<jsdl:userName>administrator</jsdl:userName>
<jsdl:password>password</jsdl:password>
</jsdlws:credentials>
</jsdlws:wsToInvoke>
</jsdlws:ws>
</jsdl:application>
<jsdl:resources>
<jsdl:candidateHosts>
<jsdl:hostName>${host}</jsdl:hostName>
</jsdl:candidateHosts>
</jsdl:resources>
</jsdl:jobDefinition>
```

O exemplo a seguir se aplica se você usar uma conexão HTTPS com o agente que executa a tarefa de serviços da Web. Mostra como configurar a chave `JVMOptions` no arquivo `jobManager.ini` do agente para apontar para os certificados de segurança.

```
JVMOptions=-Djavax.net.ssl.keyStore=/images/ITAuser/TWA/TWS/JavaExt/cfg/agentKeystore.jks
-Djavax.net.ssl.keyStorePassword=tdwb8nxt
-Djavax.net.ssl.trustStore=/images/ITAuser/TWA/TWS/JavaExt/cfg/agentKeystore.jks
-Djavax.net.ssl.trustStorePassword=tdwb8nxt
-Djavax.net.ssl.trustStoreType=JKS
```

Por padrão, o período de tempo limite para tarefas de serviços da web é de 90 segundos. Se o serviço da web não foi concluído com êxito dentro do período de tempo limite, a tarefa do Tivoli Workload Scheduler associada a ele termina com erro. Para evitar que as tarefas falhem quando a conclusão dos serviços da web demorar mais de 90 segundos, você poderá customizar o tempo limite.

O período de tempo limite é especificado no arquivo `WSJobExecutor.properties` que está localizado no diretório `TWA_HOME/TWS/JavaExt/cfg` e é estruturado como a seguir:

```
# Tempo limite de serviço da web
# Especifique o tempo limite em segundos
# O valor padrão é de 90 segundos
TIMEOUT=90
```

Por exemplo, para um serviço da web que demora 5 minutos para concluir, customize o arquivo de configuração especificando: `TIMEOUT=300`.

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

o Guia do Usuário do Console do Dynamic Workload, seção Criando Definições de Tarefa.

Para obter informações adicionais sobre como criar e editar objetos de planejamento, consulte:

o Guia do Usuário do Console do Dynamic Workload, seção Projetando sua Carga de Trabalho.

Definição de Tarefa - Tarefas de Transferência de Arquivos

Esta seção descreve os atributos necessários e opcionais para tarefas de transferência de arquivos. Apenas transferências de arquivos únicas são permitidas. Cada definição de tarefa tem o seguinte formato e argumentos:

Tabela 32. Atributos Necessários e Opcionais para a Definição de uma Tarefa de Transferência de Arquivos

Atributo	Descrição/valor	Exigido
nome do aplicativo	filetransfer	✓
Tipo de transferência de arquivos (upload ou download)	Coloque todos os atributos de transferência de arquivos entre as tags `jsdlfiletransfer:uploadInfo` ou `jsdlfiletransfer:downloadInfo` , conforme mostrado no exemplo.	✓
servidor	O endereço do servidor de transferência de arquivos (e o número da porta, se diferente da porta padrão, ao escolher FTP como o protocolo).	✓
localfile e remotefile	• Se estiver fazendo upload, `localfile` é o caminho completo e o nome do arquivo a ser transferido por upload, enquanto que `remotefile` é o caminho completo e o nome do arquivo a ser criado no destino remoto. • Se estiver fazendo download, `localfile` é o caminho completo e o nome do arquivo a ser criado no destino local, enquanto que `remotefile` é o caminho completo e o nome do arquivo a ser transferido por download. • Caracteres curinga não são suportados.	✓

localCredentials e remoteCredentials	Os nomes e senhas dos usuários autorizados nos sistemas local e remoto. Como uma alternativa aos valores reais de codificação permanente, é possível parametrizar de uma das seguintes maneiras:
	• Insira um *nome de usuário* especificado no banco de dados com a definição de usuário (aplicável a todos os sistemas operacionais nesse tipo de tarefa) e insira a instrução:
	`<jsdl:password>${password:username}</jsdl:password>`
	A senha é recuperada da definição de usuário do *nome de usuário* no banco de dados e resolvida em tempo de execução. Consulte "Utilizando Definições do Usuário sobre Tipos de Tarefas com Opções Avançadas" na página 195 para obter mais detalhes.
	Também é possível especificar o usuário de uma estação de trabalho diferente e usar a seguinte sintaxe para a senha:
	`<jsdl:password>${password:workstation#username}</jsdl:password>`
	• Insira um usuário e senha definidos com o comando do utilitário `param` localmente no agente dinâmico que executará a tarefa (se a tarefa for enviada para um conjunto ou para um conjunto dinâmico, a definição deve estar presente em todos os agentes do conjunto). Desde que você tenha definido o nome do usuário com a variável *user* e uma senha, as instruções da credencial correspondente seriam:
	`<jsdl:userName>${agent:user}</jsdl:userName>`
	`<jsdl:password>${agent:password.user}</jsdl:password>`
	As variáveis usuário e senha são resolvidas no agente no tempo de execução. Consulte "Definindo Variáveis e Senhas para Resolução Local em Agentes Dinâmicos" na página 477 para obter mais detalhes.
		✓

Tabela 32. Atributos Necessários e Opcionais para a Definição de uma Tarefa de Transferência de Arquivos (continuação)

Atributo	Descrição/valor	Exigido
protocol	Pode ser: **WINDOWS** O protocolo de compartilhamento de arquivo da Microsoft. Se você não especificar um protocolo e SSH não funcionar, WINDOWS é assumido. Especifique o diretório compartilhado na palavra-chave `remotefile`, sem especificar quaisquer caminhos em que o diretório compartilhado está aninhado. Especifique o endereço da estação de trabalho hosting o diretório compartilhado na palavra-chave `server`. O diretório ativo padrão é: `C:\Program Files\IBM\TWA\TWS\ITA`. **SSH** Um protocolo de rede que fornece funções de acesso a arquivo, transferência de arquivos e gerenciamento de arquivos sobre qualquer fluxo de dados confiável. Se você não especificar um protocolo, SSH é assumido como padrão. **FTP** Um protocolo de rede padrão usado para trocar arquivos em uma rede baseada em TCP/IP, como a Internet. **FTPS** Uma extensão para o Protocolo de Transferência de Arquivos (FTP) que inclui suporte ao protocolo criptográfico TLS (Tranport Layer Security). Especificamente, a transferência de arquivos é executada usando implicitamente o protocolo de segurança TLS para as sessões FTP, fornecendo um nível de segurança privado para a conexão de dados. A versão 1 de protocolo TLS é suportada. A configuração de reutilização da sessão SSL não é suportada. **FTPES** Uma extensão para o Protocolo de Transferência de Arquivos (FTP) que inclui suporte ao protocolo criptográfico TLS (Tranport Layer Security). Especificamente, a transferência de arquivos é executada usando explicitamente o protocolo de segurança TLS para as sessões FTP, fornecendo um nível de segurança privado para a conexão de dados. A versão 1 de protocolo TLS é suportada. A configuração de reutilização da sessão SSL não é suportada.	
transferMode	Pode ser `binary` (o padrão) ou `ascii`.	

remoteCodepage	A página de códigos usada na estação de trabalho remota. Se desejar usar uma página de código customizado, defina o parâmetro `remoteCodepage` como a seguir:
	`<jsdlfiletransfer:remoteCodepage>USER:`*`CUSTOM_CP`*
	`</jsdlfiletransfer:remoteCodepage>`
	
	em que *CUSTOM_CP* é a página de código definida pelo usuário.
	
	Por exemplo, para usar a página de código customizado `tcpip.ftpd.ftpxlbin.frence3`, defina o parâmetro `remoteCodepage` como a seguir:
	`<jsdlfiletransfer:remoteCodepage>USER:tcpip.ftpd.ftpxlbin.frence3`
	`</jsdlfiletransfer:remoteCodepage>`
		Necessário se você especificar localCodepage

localCodepage	A página de códigos usada na estação de trabalho local.	Necessário se você especificar remoteCodepage
Tempo Limite	Especifica o número de segundos a ser usado para a operação de transferência de arquivos. O valor padrão é 60 segundos.	

portsRange	Quando o modo ativo está ativado, a seção `portsRange` restringe os números de porta enviados pelo comando FTP PORT. Essa opção acomoda regras de firewall altamente restritivas. A seção `portsRange` define o intervalo de portas a ser usado no lado do cliente das conexões de dados TCP. Se você não especificar a seção `portsRange`, o sistema operacional determina os números de porta a serem usados.
	
	O parâmetro `portsRange` requer os seguintes parâmetros:
	
	min *(min_port)*
	O valor de porta mínimo a ser usado no lado do cliente das conexões de dados TCP. Os valores permitidos variam de 0 a 65535. Por exemplo, se você configurar esse valor para 1035, o Tivoli Workload Scheduler restringirá os números de porta para serem maiores ou iguais à porta 1035
	
	max *(max_port)*
	O valor de porta máximo a ser usado no lado do cliente das conexões de dados TCP. Os valores permitidos variam de 0 a 65535. Por exemplo, se você configurar esse valor para 1038, o Tivoli Workload Scheduler restringe o número de porta para menor ou igual à porta 1038.
	
	Por exemplo, para limitar o intervalo de portas a ser usado no lado do cliente entre a porta 1035 e a porta 1040, especifique o seguinte:
	`<jsdlfiletransfer:portsRange>`
	`<jsdlfiletransfer:min>1035</jsdlfiletransfer:min>`
	`<jsdlfiletransfer:max>1040</jsdlfiletransfer:max>`
	`</jsdlfiletransfer:portsRange>`	

Tabela 32. Atributos Necessários e Opcionais para a Definição de uma Tarefa de Transferência de Arquivos (continuação)

Atributo	Descrição/valor	Exigido
passiveMode	Especifica se o servidor é passivo ou ativo no estabelecimento das conexões para transferências de dados. Se você definir essa opção para NÃO, o servidor estabelecerá a conexão de dados com o cliente (modo ativo). Se você definir essa opção para SIM, o cliente estabelecerá a conexão de dados com o servidor (modo passivo). O valor padrão é NÃO.	

O seguinte arquivo xml mostra a seção "aplicativo" JSDL de uma definição de tarefa de amostra para um tipo de tarefa de transferência de arquivos:

```
<jsdl:application name="filetransfer">
<jsdlfiletransfer:filetransfer>
<jsdlfiletransfer:downloadInfo>
<jsdlfiletransfer:server>FTP_SERVER</jsdlfiletransfer:server>
<jsdlfiletransfer:localfile>LOCAL_FILE</jsdlfiletransfer:localfile>
<jsdlfiletransfer:remotefile>REMOTE_FILE</jsdlfiletransfer:remotefile>
<jsdlfiletransfer:remoteCredentials>
<jsdl:userName>USERNAME</jsdl:userName>
<jsdl:password>PASSWORD</jsdl:password>
</jsdlfiletransfer:remoteCredentials>
<jsdlfiletransfer:protocol>PROTOCOL</jsdlfiletransfer:protocol>
<jsdlfiletransfer:transferMode>ASCII_BINARY</jsdlfiletransfer:transferMode>
<jsdlfiletransfer:codepageConversion>
<jsdlfiletransfer:remoteCodepage>RM_CP</jsdlfiletransfer:remoteCodepage>
<jsdlfiletransfer:localCodepage>LC_CP</jsdlfiletransfer:localCodepage>
</jsdlfiletransfer:codepageConversion>
<jsdlfiletransfer:timeout>CONNECTION_TIMEOUT</jsdlfiletransfer:timeout>
<jsdlfiletransfer:portsRange>
<jsdlfiletransfer:min>MIN_PORT</jsdlfiletransfer:min>
<jsdlfiletransfer:max>MAX_PORT</jsdlfiletransfer:max>
</jsdlfiletransfer:portsRange>
<jsdlfiletransfer:passiveMode>YES_NO</jsdlfiletransfer:passiveMode>
</jsdlfiletransfer:downloadInfo>
</jsdlfiletransfer:filetransfer>
</jsdl:application>
```

O exemplo a seguir mostra uma tarefa generalizada que faz download de um arquivo de uma estação de trabalho remota com endereço 10.0.0.8:

```
$JOBS
AGENT#FILE_TRANSFER
TASK
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
 xmlns:jsdlfiletransfer="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdlfiletransfer"
name="FILETRANSFER">
<jsdl:application name="filetransfer">
<jsdlfiletransfer:filetransfer>
<jsdlfiletransfer:downloadInfo>
<jsdlfiletransfer:server>10.0.0.8</jsdlfiletransfer:server>
<jsdlfiletransfer:localfile>c:\MyTextFile.txt</jsdlfiletransfer:localfile>
<jsdlfiletransfer:remotefile>./MyRemoteFile.txt</jsdlfiletransfer:remotefile>
<jsdlfiletransfer:localCredentials>
<jsdl:userName>${agent:locluser}</jsdl:userName>
<jsdl:password>${agent:password.${agent:locluser}}</jsdl:password>[1]
</jsdlfiletransfer:localCredentials>
<jsdlfiletransfer:remoteCredentials>
<jsdl:userName>remuser</jsdl:userName>
<jsdl:password>${password:remuser}</jsdl:password>[2]
</jsdlfiletransfer:remoteCredentials>
<jsdlfiletransfer:protocol>FTP</jsdlfiletransfer:protocol>
<jsdlfiletransfer:transferMode>ascii</jsdlfiletransfer:transferMode>
<jsdlfiletransfer:codepageConversion>
<jsdlfiletransfer:remoteCodepage>IBM-280</jsdlfiletransfer:remoteCodepage>
<jsdlfiletransfer:localCodepage>ISO8859-1</jsdlfiletransfer:localCodepage>
</jsdlfiletransfer:codepageConversion>
</jsdlfiletransfer:downloadInfo>
</jsdlfiletransfer:filetransfer>
</jsdl:application>
</jsdl:jobDefinition>
```

Nota: nas seções de credenciais,

1. O nome do usuário local foi definido no agente que executa a tarefa com uma variável nomeada `locluser` por meio do comando do utilitário `param`. Assim, o valor definido para `locluser` será recuperado no tempo de execução do arquivo de variáveis localizado no agente. Da mesma forma, a senha para o valor

representada por `locluser` foi definida no agente com o comando `param` e será resolvida no tempo de execução a partir do arquivo de variáveis locais.

2. O nome do usuário remoto foi definido com o comando `composer user` e está armazenado no banco de dados junto com sua senha como nome de usuário `remuser`. A senha para `remuser` será recuperada do banco de dados no tempo de execução.

O exemplo a seguir mostra uma definição de tarefa a ser usada para efetuar o download de um arquivo de texto usando o protocolo FTPES com um modo ativo, um tempo limite de 10 segundos e um intervalo de porta a usar entre 1035 e 1038:

```
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdlfiletransfer="http://www.ibm.com/xmlns/prod/scheduling/1.0/
jsdlfiletransfer" name="FTPES_DOWNLOAD_TEXT">
<jsdl:application name="filetransfer">
<jsdlfiletransfer:filetransfer>
<jsdlfiletransfer:downloadInfo>
<jsdlfiletransfer:server>myServerFtp</jsdlfiletransfer:server>
<jsdlfiletransfer:localfile>d:\MyLocalFile.txt</jsdlfiletransfer:localfile>
<jsdlfiletransfer:remotefile>/tmp/MyRemoteFile.txt</jsdlfiletransfer:remotefile>
<jsdlfiletransfer:remoteCredentials>
<jsdl:userName>myUser</jsdl:userName>
<jsdl:password>myPassword</jsdl:password>
</jsdlfiletransfer:remoteCredentials>
<jsdlfiletransfer:protocol>FTPES</jsdlfiletransfer:protocol>
<jsdlfiletransfer:transferMode>ascii</jsdlfiletransfer:transferMode>
<jsdlfiletransfer:timeout>10</jsdlfiletransfer:timeout>
<jsdlfiletransfer:portsRange>
<jsdlfiletransfer:min>1035</jsdlfiletransfer:min>
<jsdlfiletransfer:max>1038</jsdlfiletransfer:max>
</jsdlfiletransfer:portsRange>
</jsdlfiletransfer:downloadInfo>
</jsdlfiletransfer:filetransfer>
</jsdl:application>
</jsdl:jobDefinition>
```

O exemplo a seguir mostra uma definição de tarefa a ser usada para transferir um arquivo de texto usando o protocolo SSH:

```
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdlfiletransfer="http://www.ibm.com/xmlns/prod/scheduling/1.0/
jsdlfiletransfer" name="SSH_UPLOAD">
<jsdl:application name="filetransfer">
<jsdlfiletransfer:filetransfer>
<jsdlfiletransfer:uploadInfo>
<jsdlfiletransfer:server>myServer</jsdlfiletransfer:server>
<jsdlfiletransfer:localfile>d:\MyLocalFile.txt</jsdlfiletransfer:localfile>
<jsdlfiletransfer:remotefile>/tmp/MyRemoteFile.txt</jsdlfiletransfer:remotefile>
<jsdlfiletransfer:remoteCredentials>
<jsdl:userName>myUser</jsdl:userName>
<jsdl:password>myPassword</jsdl:password>
</jsdlfiletransfer:remoteCredentials>
<jsdlfiletransfer:protocol>SSH</jsdlfiletransfer:protocol>
<jsdlfiletransfer:transferMode>ascii</jsdlfiletransfer:transferMode>
</jsdlfiletransfer:uploadInfo>
</jsdlfiletransfer:filetransfer>
</jsdl:application>
</jsdl:jobDefinition>
```

O exemplo a seguir mostra uma definição de tarefa a ser usada para transferir um arquivo de texto usando o protocolo WINDOWS:

```
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdlfiletransfer="http://www.ibm.com/xmlns/prod/scheduling/1.0/
jsdlfiletransfer" name="WINDOWS_DOWNLOAD">
<jsdl:application name="filetransfer">
<jsdlfiletransfer:filetransfer>
<jsdlfiletransfer:downloadInfo>
<jsdlfiletransfer:server>myServer</jsdlfiletransfer:server>
<jsdlfiletransfer:localfile>d:\MyLocalFile.txt</jsdlfiletransfer:localfile>
<jsdlfiletransfer:remotefile>mySharedFolder\MyRemoteFile.txt
</jsdlfiletransfer:remotefile>
<jsdlfiletransfer:remoteCredentials>
<jsdl:userName>myUser</jsdl:userName>
<jsdl:password>myPassword</jsdl:password>
</jsdlfiletransfer:remoteCredentials>
<jsdlfiletransfer:protocol>WINDOWS</jsdlfiletransfer:protocol>
<jsdlfiletransfer:transferMode>ascii</jsdlfiletransfer:transferMode>
</jsdlfiletransfer:downloadInfo>
</jsdlfiletransfer:filetransfer>
</jsdl:application>
</jsdl:jobDefinition>
```

O tipo de tarefa de transferência de arquivos fornece os seguintes códigos de retorno. Códigos de retorno estão disponíveis somente para tarefas concluídas.

RC=0 Transferência de arquivos concluída com êxito.

RC=-1 Transferência de arquivos não executada. A tarefa falhou com o seguinte código de erro:

AWKFTE007E

Explicação: Ocorreu um erro durante a operação de transferência de arquivos.

Possíveis razões: Arquivo remoto não localizado ou permissão negada.

RC=-2 Transferência de arquivos não executada. A tarefa falhou com o seguinte código de erro:

AWKFTE020E

Explicação: Apenas para protocolos SSH ou WINDOWS. Um erro foi retornado durante tentativa de converter a página de códigos.

Possíveis razões: Para protocolos SSH ou WINDOWS, a página de códigos é automaticamente detectada e convertida. Nesse caso, há um erro na página de códigos do arquivo a ser transferido, que não está em conformidade com a página de códigos do sistema local.

RC=-3 Transferência de arquivos não executada. A tarefa falhou com o seguinte código de erro:

WKFTE015E

Explicação: Ocorreu um erro durante a operação de transferência de arquivos.

Possíveis razões: Arquivo local não localizado.

RC=-4 Transferência de arquivos executada com a página de códigos padrão. A tarefa falhou com o seguinte código de erro:

AWKFTE023E

Explicação: A conversão de página de código especificada não foi executada. A transferência de arquivos foi executada com páginas de código padrão.

Possível razão: A página de código especificada está indisponível.

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

o Guia do Usuário do Console do Dynamic Workload, seção Criando Definições de Tarefa.

Para obter informações adicionais sobre como criar e editar objetos de planejamento, consulte:

o Guia do Usuário do Console do Dynamic Workload, seção Projetando sua Carga de Trabalho.

Definição de Tarefa - Tarefas de Provisioning

Você deve ter o IBM SmartCloud Provisioning versão 2.1 para poder usar o tipo de tarefa do Provisioning no Tivoli Workload Scheduler.

Para ler um cenário da vida real comum que atinge metas de negócios, incluindo a implementação de uma tarefa do Provisioning, consulte http://pic.dhe.ibm.com/infocenter/tivihelp/v47r1/index.jsp?topic=/com.ibm.tivoli.itws.doc_9.2/scen/awsbsscp_rc.html.

Antes de definir uma tarefa do Provisioning, você deve ter armazenado o certificado do servidor HTTP do Provisioning e definido-o, conforme explicado em "Etapas de Pré-requisitos para Criar Tarefas do Provisioning" no *Tivoli Workload Automation: Dynamic Workload Console User's Guide*.

Esta seção descreve os atributos necessários e opcionais para tarefas de Provisioning. Cada definição de tarefa tem o seguinte formato e argumentos:

Tabela 33. Atributos necessários e opcionais para a definição de uma tarefa de Provisioning

Atributo	Descrição/valor	Exigido
nome do aplicativo	provisioning	✔
actionType	Valores válidos: **deploy** Implementar uma imagem virtual no grupo de nuvens e criar uma nova instância do sistema virtual contendo o número de instâncias de imagem virtual especificadas. **manage** Iniciar, parar ou excluir recursos virtuais (como máquinas virtuais ou imagens virtuais de ambiente complexo).	✔
cloudGroupId	O identificador exclusivo do grupo de nuvens do qual está sendo escolhida a instância virtual a ser implementada.	✔
instanceId	O identificador exclusivo da instância virtual existente na qual você está agindo. Uma instância virtual pode ser uma ou mais máquinas virtuais. Se desejar agir em uma única máquina virtual pertencente à instância atual, especifique seu ID como o atributo **virtualMachineId**.	✔
virtualMachineId	O identificador exclusivo da máquina virtual existente na qual você está agindo.	
VirtualImageId	O identificador exclusivo da imagem virtual existente que é usada como um modelo para uma nova imagem virtual que está sendo implementada.	✔
instanceNameDeploy	O nome exclusivo da instância virtual.	✔
numberOfVirtualMachines	O número de máquinas virtuais que estão sendo implementadas.	✔
descrição	Uma sequência de texto que descreve a instância de sistema virtual.	
tags	Uma matriz de tags definida pelo usuário. Estes dados do usuário especificados aqui podem ser recuperados dentro da máquina virtual. Eles podem ser usados para configurar uma máquina virtual na primeira inicialização ou para executar um script de inicialização registrado na máquina virtual.	
Tamanho	O tamanho da instância. Este é o tamanho de cada máquina virtual única pertencente à instância. Valores válidos: • xsmall: Extra pequeno • small: Pequeno • medium: Médio • large: Grande • xlarge: Extra Grande	✔
winPassword	A senha do administrador para acessar os sistemas Windows implementados.	
unixSSHPublicKey	Aplicável apenas ao UNIX. A chave de Shell Seguro deve ser fornecida pelo administrador do Provisioning.	

Tabela 33. Atributos necessários e opcionais para a definição de uma tarefa de Provisioning (continuação)

Atributo	Descrição/valor	Exigido

userName
 	
password
	As credenciais associadas ao servidor Provisioning. Como uma alternativa aos valores reais de codificação permanente, é possível parametrizar de uma das seguintes maneiras:
	• Insira um *nome de usuário* especificado no banco de dados com a definição de usuário (aplicável a todos os sistemas operacionais nesse tipo de tarefa) e insira a instrução:
	`<jsdl:password>${password:username}</jsdl:password>`
	A senha é recuperada da definição de usuário do *nome de usuário* no banco de dados e resolvida em tempo de execução. Consulte "Utilizando Definições do Usuário sobre Tipos de Tarefas com Opções Avançadas" na página 195 para obter mais detalhes.
	Também é possível especificar o usuário de uma estação de trabalho diferente e usar a seguinte sintaxe para a senha:
	`<jsdl:password>${password:workstation#username}</jsdl:password>`
	• Insira um usuário e senha definidos com o comando do utilitário `param` localmente no agente dinâmico que executará a tarefa (se a tarefa for enviada para um conjunto ou para um conjunto dinâmico, a definição deve estar presente em todos os agentes do conjunto). Desde que você tenha definido o nome do usuário com a variável *user* e uma senha, as instruções da credencial correspondente seriam:
	`<jsdl:userName>${agent:user}</jsdl:userName>`
	`<jsdl:password>${agent:password.user}</jsdl:password>`
	As variáveis usuário e senha são resolvidas no agente no tempo de execução. Consulte "Definindo Variáveis e Senhas para Resolução Local em Agentes Dinâmicos" na página 477 para obter mais detalhes.
		✓

hostname	O nome do host do servidor Provisioning.	✓
porta	O número da porta do servidor Provisioning. O valor padrão é `443`.	✓
manageType	Valores válidos: • **actionStart**: Inicia a instância de sistema virtual. • **actionStop**: Para a instância de sistema virtual. • **actionDelete**: Exclui a instância de sistema virtual.	✓

O exemplo a seguir mostra uma definição de tarefa a ser usada para implementar uma máquina virtual no ambiente do Provisioning:

```
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl" xmlns:
   jsdlprovisioning="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdlprovisioning"
          name="PROVISIONING">
  <jsdl:application name="provisioning">
    <jsdlprovisioning:provisioning>
   <jsdlprovisioning:ProvisioningParameters>
    <jsdlprovisioning:actions>
     <jsdlprovisioning:actionType>
      <jsdlprovisioning:deploy>
       <jsdlprovisioning:groupCloud>
        <jsdlprovisioning:cloudGroupId>8</jsdlprovisioning:cloudGroupId>
       </jsdlprovisioning:groupCloud>
       <jsdlprovisioning:groupVirtualImage>
        <jsdlprovisioning:virtualImageId>52</jsdlprovisioning:virtualImageId>
       </jsdlprovisioning:groupVirtualImage>
       <jsdlprovisioning:instanceNameDeploy>TESTPROP</jsdlprovisioning:instanceNameDeploy>
       <jsdlprovisioning:numberOfVirtualMachines>1</jsdlprovisioning:numberOfVirtualMachines>
       <jsdlprovisioning:description/>
       <jsdlprovisioning:tags/>
       <jsdlprovisioning:size>xsmall</jsdlprovisioning:size>
       <jsdlprovisioning:winPassword/>
       <jsdlprovisioning:unixSSHPublicKey/>
      </jsdlprovisioning:deploy>
     </jsdlprovisioning:actionType>
    </jsdlprovisioning:actions>
    <jsdlprovisioning:connectionInfo>
     <jsdlprovisioning:credentials>
      <jsdl:userName>cbadmin</jsdl:userName>
      <jsdl:password>{aes}2WfJH/3aOxyX2f+QXeW+1YnrN2tM4z338QMYlYVgpOA=</jsdl:password>
     </jsdlprovisioning:credentials>
     <jsdlprovisioning:server>
      <jsdlprovisioning:hostname>9.168.58.192</jsdlprovisioning:hostname>
      <jsdlprovisioning:port>443</jsdlprovisioning:port>
     </jsdlprovisioning:server>
    </jsdlprovisioning:connectionInfo>
```

```
   </jsdlprovisioning:ProvisioningParameters>
  </jsdlprovisioning:provisioning>
  </jsdl:application>
</jsdl:jobDefinition>
```

O exemplo a seguir mostra uma definição de tarefa a ser usada para parar uma máquina virtual:

```
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl" xmlns:
   jsdlprovisioning="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdlprovisioning"
              name="PROVISIONING">
  <jsdl:application name="provisioning">
    <jsdlprovisioning:provisioning>
   <jsdlprovisioning:ProvisioningParameters>
    <jsdlprovisioning:actions>
     <jsdlprovisioning:actionType>
      <jsdlprovisioning:manage>
       <jsdlprovisioning:manageType>actionStop</jsdlprovisioning:manageType>
       <jsdlprovisioning:instanceId>102</jsdlprovisioning:instanceId>
       <jsdlprovisioning:virtualMachineId/>
      </jsdlprovisioning:manage>
     </jsdlprovisioning:actionType>
    </jsdlprovisioning:actions>
    <jsdlprovisioning:connectionInfo>
     <jsdlprovisioning:credentials>
      <jsdl:userName>cbadmin</jsdl:userName>
      <jsdl:password>{aes}2WfJH/3aOxyX2f+QXeW+1YnrN2tM4z338QMYlYVgpOA=</jsdl:password>
     </jsdlprovisioning:credentials>
     <jsdlprovisioning:server>
      <jsdlprovisioning:hostname>9.168.58.192</jsdlprovisioning:hostname>
      <jsdlprovisioning:port>443</jsdlprovisioning:port>
     </jsdlprovisioning:server>
    </jsdlprovisioning:connectionInfo>
   </jsdlprovisioning:ProvisioningParameters>
  </jsdlprovisioning:provisioning>
  </jsdl:application>
</jsdl:jobDefinition>
```

Você planeja tarefas do Tivoli Workload Scheduler Provisioning definindo-as em fluxos de tarefas. Inclua a tarefa em um fluxo de tarefas com todos os argumentos de planejamento necessários e envie-a. As tarefas podem ser enviadas em um ambiente distribuído usando o Dynamic Workload Console ou a linha de comandos **conman**. As tarefas podem ser enviadas em um ambiente z/OS, usando o Dynamic Workload Console ou o aplicativo ISPF. Após o envio, quando a tarefa estiver em execução (status **EXEC**), será possível executar **kill** na tarefa, se necessário. No entanto, esta ação é efetiva com relação à tarefa do Tivoli Workload Scheduler, mas ela não afeta o comando ativado remotamente. Após uma ação **kill**, o Tivoli Workload Scheduler coletará o log da tarefa quando o agente for reiniciado e designará o status **Error** ou **ABEND** para a tarefa Tivoli Workload Scheduler, independentemente do status da tarefa Provisioning.

Se o agente Tivoli Workload Scheduler se tornar indisponível durante o envio da tarefa do Tivoli Workload Scheduler Provisioning ou durante a execução da tarefa, quando o agente se tornar disponível novamente, o Tivoli Workload Scheduler iniciará o monitoramento da tarefa a partir de onde ele parou.

Para obter informações adicionais sobre o Provisioning, consulte o Centro de Informações do IBM SmartCloud Provisioning

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

o Guia do Usuário do Console do Dynamic Workload, seção Criando Definições de Tarefa.

Para obter informações adicionais sobre como criar e editar objetos de planejamento, consulte:

o Guia do Usuário do Console do Dynamic Workload, seção Projetando sua Carga de Trabalho.

Definição de Tarefa - Tarefas J2EE

Esta seção descreve os atributos necessários e opcionais para tarefas de serviços da Web. Cada definição de tarefa tem o seguinte formato e argumentos:

Tabela 34. Atributos Obrigatórios e Opcionais para a Definição de uma Tarefa J2EE.

Atributo	Descrição/valor	Exigido
nome do aplicativo	j2ee	✓
jms operation	A operação a ser executada. Os valores suportados são: • **send**. Este é o valor padrão. • **receive**. Se você especificar **receive**, é possível definir como opção um valor para o atributo **timeout**.	
timeout	O tempo limite, expressado em segundos, dentro dos quais a tarefa deve ser concluída. Se você não especificar um tempo limite ou configurá-lo para 0, a tarefa continua indefinidamente.	
connectionURL	A URL do WebSphere Application Server.	
connFactory	Um objeto administrado que um cliente usa para criar uma conexão com o provedor JMS. Para especificar o connection factory, é possível usar uma expressão de variável que pode conter uma ou mais referências de variáveis, como ${var}, qualquer caractere e qualquer sequência.	✓
destino	Um objeto administrado que encapsula a identidade de um destino de mensagem, que é onde as mensagens são entregues e consumidas. Para especificar o destino, é possível usar uma expressão de variável que pode conter uma ou mais referências de variáveis, como ${var}, qualquer caractere e qualquer sequência.	✓
message	A mensagem a ser enviada.	✓ **Nota:** Esse atributo é necessário somente para a tarefa de envio.

Credenciais	Especifique o nome e senha de usuário a serem usados ao executar o aplicativo J2EE. Use este campo se a segurança global estiver ativada no WebSphere Application Server. O usuário deve ser definido em WebSphere Application Server. Para especificar essas credenciais, é possível usar expressões de variável que possam conter uma ou mais referências de variáveis, como ${var}, opcionalmente em associação com qualquer caractere ou com uma sequência simples. Além disso, é possível parametrizar de uma das seguintes maneiras:
	• Insira um *nome de usuário* especificado no banco de dados com a definição de usuário de *nome de usuário* (é aplicável a todos os sistemas operacionais nesse tipo de tarefa) e insira a instrução:
	`<jsdl:password>${password:username}</jsdl:password>`
	A senha é recuperada da definição de usuário do *nome de usuário* no banco de dados e resolvida em tempo de execução. Consulte "Utilizando Definições do Usuário sobre Tipos de Tarefas com Opções Avançadas" na página 195 para obter mais detalhes.
	Também é possível especificar o usuário de uma estação de trabalho diferente e usar a seguinte sintaxe para a senha:
	`<jsdl:password>${password:workstation#username}</jsdl:password>`
	• Insira um usuário e senha definidos com o comando do utilitário `param` localmente no agente dinâmico que executará a tarefa (se a tarefa for enviada para um conjunto ou para um conjunto dinâmico, a definição deve estar presente em todos os agentes do conjunto). Desde que você tenha definido o nome do usuário com a variável *user* e uma senha, as instruções da credencial correspondente seriam:
	`<jsdl:userName>${agent:user}</jsdl:userName>`
	`<jsdl:password>${agent:password.user}</jsdl:password>`
	As variáveis usuário e senha são resolvidas no agente no tempo de execução. Consulte "Definindo Variáveis e Senhas para Resolução Local em Agentes Dinâmicos" na página 477 para obter mais detalhes.	

O exemplo a seguir mostra uma tarefa de envio que envia uma mensagem à fila MyQueue:

```
$JOBS
AGENT#JOB_NAME_JMS_SEND
TASK
```

```
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
 xmlns:jsdlj="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdlj" name="JMS_JOB_SEND">
<jsdl:application name="j2ee">
<jsdlj:j2ee>
<jsdlj:jms operation="send">
<jsdlj:connectionURL>corbaloc:iiop:washost.mydomain.com:2809</jsdlj:connectionURL>
<jsdlj:connFactory>jms/MyCF</jsdlj:connFactory>
<jsdlj:destination>jms/MyQueue</jsdlj:destination>
<jsdlj:message>Submission of jms job: SEND MESSAGE</jsdlj:message>
</jsdlj:jms>
<jsdlj:credentials>
<jsdlj:userName>jtwoeeuser</jsdlj:userName>
<jsdlj:password>${password:jtwoeeuser}</jsdlj:password>[1]
</jsdlj:credentials>
</jsdlj:j2ee>
</jsdl:application>
</jsdl:jobDefinition>
```

Nota: (1) O usuário `jtwoeeuser` foi definido no banco de dados do Tivoli Workload Scheduler usando o comando de definição de usuário *nome de usuário*. A senha associada, especificada pela sequência `${password:jtwoeeuser}` na tarefa, será recuperada do banco de dados no tempo de execução.

O exemplo a seguir mostra uma tarefa que lê mensagens da fila MyQueue:

```
$JOBS
AGENT#JOB_NAME_JMS_RECEIVE
TASK
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
 xmlns:jsdlj="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdlj" name="JMS_JOB_RECEIVE">
<jsdl:application name="j2ee">
<jsdlj:j2ee>
<jsdlj:jms operation="receive" timeout="180">
<jsdlj:connFactory>jms/MyCF</jsdlj:connFactory>
<jsdlj:destination>jms/MyQueue</jsdlj:destination>
</jsdlj:jms>
<jsdlj:credentials>
<jsdlj:userName>userName</jsdlj:userName>
<jsdlj:password>password</jsdlj:password>
</jsdlj:credentials>
</jsdlj:j2ee>
</jsdl:application>
</jsdl:jobDefinition>
```

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

o Guia do Usuário do Console do Dynamic Workload, seção Criando Definições de Tarefa.

Para obter informações adicionais sobre como criar e editar objetos de planejamento, consulte:

o Guia do Usuário do Console do Dynamic Workload, seção Projetando sua Carga de Trabalho.

Definição de Tarefa - Tarefas de Banco de Dados

Esta seção descreve os atributos necessários e opcionais para tarefas de banco de dados. Cada definição de tarefa tem o seguinte formato e argumentos:

Tabela 35. Atributos Necessários e Opcionais para a Definição de uma Tarefa de Banco de Dados

Atributo	Descrição/valor	Exigido
nome do aplicativo	banco de dados	✓

Tabela 35. Atributos Necessários e Opcionais para a Definição de uma Tarefa de Banco de Dados (continuação)

Atributo	Descrição/valor	Exigido
dbms	O tipo de banco de dados onde você deseja que a tarefa seja executada. Os valores suportados são: **db2** Para os bancos de dados DB2 **mssql** Para os bancos de dados Microsoft SQL Server **oracle** Para os bancos de dados Oracle	✓
servidor	O nome do host do servidor onde o banco de dados está localizado.	✓
porta	O número da porta para a tarefa de banco de dados.	✓
banco de dados	O nome do banco de dados do .	✓
JDBC driver class name	O nome da classe do driver JDBC	Necessário se você especificar um banco de dados customizado.
JDBC connection string	A sequência que é usada para conectar ao banco de dados, contendo a URL do banco de dados, o nome de usuário e a senha	Necessário se você especificar um banco de dados customizado.
JDBC jar class path	Caminho para os arquivos jar do cliente de banco de dados. Esse valor substitui o valor especificado no arquivo de configuração DatabaseJobExecutor.properties, se houver algum. Se você selecionar o banco de dados Microsoft SQL Server, a versão 4 dos drivers JDBC é necessária.	
dbStatement	O nome da tarefa de banco de dados a ser executada.	Necessário se você especificar uma instrução SQL

Nome de storedProcedure	O nome do procedimento armazenado nos bancos de dados DB2, Oracle ou MSSQL.
	O procedimento não poderá ser armazenado no DB2 se o banco de dados já contiver um ou mais procedimentos armazenados com o mesmo nome e esquema. Por exemplo, se o banco de dados tiver mais de um procedimento armazenado, denominado `TEST.STORE_PROC1`, com diferentes parâmetros como no seguinte:
	`TEST.STORE_PROC1(VARCHAR,?)`
	`TEST.STORE_PROC1(VARCHAR,VARCHAR,?)`
	`TEST.STORE_PROC1(VARCHAR,?,?)`
	`TEST.STORE_PROC1(VARCHAR,VARCHAR,?,?)`
	então, a tarefa do banco de dados não poderá ser criada e a seguinte mensagem será retornada: `AWKDBE033E O nome do procedimento armazenado fornecido corresponde a mais de uma definição de procedimento armazenado no banco de dados, para desfazer a ambiguidade, especifique também o esquema.`
		Necessário se você especificar um procedimento armazenado.

Chave ParameterTableValue	O nome e valores do procedimento expressos de acordo com a seguinte sintaxe:
	Tipo de variável de procedimento armazenado
	Os valores suportados são:
	• IN
	• OUT
	• INOUT
	Nome da variável
	O nome da variável conforme definido no procedimento armazenado.
	Tipo de variável
	Tipo de variável. Os tipos SQL suportados são:
	• DATE
	• DECIMAL
	• INTEGER
	• VARCHAR
	0...n Posição de cada variável conforme definido no procedimento armazenado.
	Por exemplo:
	`Name Value`
	`IN VARIN DATE 0 2012-01-01`
	`OUT VAROUT DATE 1 ?`
		Necessário se você especificar um procedimento armazenado.

Conteúdo de ParameterTableValue	Valor da variável. Para variáveis de saída, o valor deve ser: **?**. Para inserir uma variável de data, use o seguinte formato: **aaaa-mm-dd**. Se nenhum valor for especificado para um parâmetro, o valor será considerado um valor NULO no banco de dados.	Necessário se você especificar um procedimento armazenado.

Tabela 35. Atributos Necessários e Opcionais para a Definição de uma Tarefa de Banco de Dados (continuação)

Atributo	Descrição/valor	Exigido

credenciais	O nome e a senha de usuário para acessar o banco de dados (usuários do domínio não são suportados). Como uma alternativa aos valores reais de codificação permanente, é possível parametrizar de uma das seguintes maneiras:
	• Insira um *nome de usuário* especificado no banco de dados com a definição de usuário (aplicável a todos os sistemas operacionais nesse tipo de tarefa) e insira a instrução:
	`<jsdl:password>${password:username}</jsdl:password>`
	A senha é recuperada da definição de usuário do *nome de usuário* no banco de dados e resolvida em tempo de execução. Consulte "Utilizando Definições do Usuário sobre Tipos de Tarefas com Opções Avançadas" na página 195 para obter mais detalhes.
	Também é possível especificar o usuário de uma estação de trabalho diferente e usar a seguinte sintaxe para a senha:
	`<jsdl:password>${password:workstation#username}</jsdl:password>`
	• Insira um usuário e senha definidos com o comando do utilitário `param` localmente no agente dinâmico que executará a tarefa (se a tarefa for enviada para um conjunto ou para um conjunto dinâmico, a definição deve estar presente em todos os agentes do conjunto). Desde que você tenha definido o nome do usuário com a variável *user* e uma senha, as instruções da credencial correspondente seriam:
	`<jsdl:userName>${agent:user}</jsdl:userName>`
	`<jsdl:password>${agent:password.user}</jsdl:password>`
	As variáveis usuário e senha são resolvidas no agente no tempo de execução. Consulte "Definindo Variáveis e Senhas para Resolução Local em Agentes Dinâmicos" na página 477 para obter mais detalhes.	

O exemplo a seguir mostra uma tarefa que executa uma consulta em um banco de dados DB2:

```
$JOBS
AGENT#DATABASE
TASK
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
 xmlns:jsdldatabase="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdldatabase" name="database">
  <jsdl:application name="database">
    <jsdldatabase:database>
            <jsdldatabase:sqlActionInfo>
                <jsdldatabase:dbms>db2</jsdldatabase:dbms>
                <jsdldatabase:server>localhost</jsdldatabase:server>
                <jsdldatabase:port>50000</jsdldatabase:port>
                <jsdldatabase:database>TWS32</jsdldatabase:database>
                <jsdldatabase:statements>
                    <jsdldatabase:dbStatement>SELECT * FROM DWB.ARE_ABSTRACT_
RESOURCES</jsdldatabase:dbStatement>
                </jsdldatabase:statements>
                <jsdldatabase:credentials>
                    <jsdl:userName>${agent:dbvars..dbtwouser}</jsdl:userName>
                    <jsdl:password>${agent:password.${agent:dbvars..dbtwouser}}</jsdl:password>[1]
                </jsdldatabase:credentials>
            </jsdldatabase:sqlActionInfo>
        </jsdldatabase:database>
  </jsdl:application>
</jsdl:jobDefinition>
DESCRIPTION "Definida usando o composer."
RECOVERY STOP
```

Nota: (1) o nome do usuário foi definido no agente que executa a tarefa com uma variável nomeada `dbtwouser` por meio do comando do utilitário `param`. Assim, o valor definido para `dbtwouser` será recuperado no tempo de execução do arquivo de variáveis `dbvars` localizado no agente. Da mesma forma, a senha para o valor representada por `dbtwouser` foi definida no agente com o comando `param` e será resolvida no tempo de execução a partir do mesmo arquivo de variáveis.

Definição de Tarefa - Tarefas MSSQL

Esta seção descreve os atributos necessários e opcionais para tarefas MSSQL. Cada definição de tarefa tem o seguinte formato e argumentos:

Tabela 36. Atributos Obrigatórios e Opcionais para a Definição de uma Tarefa MSSQL.

Atributo	Descrição/valor	Exigido
nome do aplicativo	banco de dados	✓
dbms	O tipo de banco de dados onde você deseja que a tarefa seja executada. Como essa tarefa é específica do banco de dados Microsoft SQL Server, o único valor suportado é mssql.	✓
JDBC jar class path	Caminho para os arquivos jar do cliente de banco de dados. Esse valor substitui o valor especificado no arquivo de configuração DatabaseJobExecutor.properties, se houver algum. A versão 4 dos drivers JDBC é necessária.	
servidor	O nome do host do servidor onde o banco de dados está localizado.	✓
porta	O número da porta para a tarefa de banco de dados.	✓
banco de dados	O nome do banco de dados do .	✓
dbStatement	A instrução SQL. Para separar instruções, use uma linha vazia.	✓

credenciais	O nome e a senha de usuário para acessar o banco de dados (usuários do domínio não são suportados). Como uma alternativa aos valores reais de codificação permanente, é possível parametrizar de uma das seguintes maneiras:
	• Insira um *nome de usuário* especificado no banco de dados com a definição de usuário de *nome de usuário* (é aplicável a todos os sistemas operacionais nesse tipo de tarefa) e insira a instrução:
	`<jsdl:password>${password:username}</jsdl:password>`
	A senha é recuperada da definição de usuário do *nome de usuário* no banco de dados e resolvida em tempo de execução. Consulte "Utilizando Definições do Usuário sobre Tipos de Tarefas com Opções Avançadas" na página 195 para obter mais detalhes.
	• Insira um usuário e senha definidos com o comando do utilitário `param` localmente no agente dinâmico que executará a tarefa (se a tarefa for enviada para um conjunto ou para um conjunto dinâmico, a definição deve estar presente em todos os agentes do conjunto). Desde que você tenha definido o nome do usuário com a variável *user* e uma senha, as instruções da credencial correspondente seriam:
	`<jsdl:userName>${agent:user}</jsdl:userName>`
	`<jsdl:password>${agent:password.user}</jsdl:password>`
	As variáveis usuário e senha são resolvidas no agente no tempo de execução. Consulte "Definindo Variáveis e Senhas para Resolução Local em Agentes Dinâmicos" na página 477 para obter mais detalhes.	

O exemplo a seguir mostra uma tarefa que executa uma tarefa em um banco de dados MSSQL:

```
$JOBS
AGENT#MSSQLJOB
TASK
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
 xmlns:jsdldatabase="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdldatabase" name="database">
  <jsdl:application name="mssqljob">
    <jsdldatabase:database>
            <jsdldatabase:sqlActionInfo>
                <jsdldatabase:dbms>mssql</jsdldatabase:dbms>
                <jsdldatabase:server>localhost</jsdldatabase:server>
                <jsdldatabase:port>111</jsdldatabase:port>
                <jsdldatabase:database>MYDATABASE</jsdldatabase:database>
                <jsdldatabase:statements>
                    <jsdldatabase:dbStatement type="job">sada</jsdldatabase:dbStatement>
                </jsdldatabase:statements>
                <jsdldatabase:credentials>
                    <jsdl:userName>mssqluser</jsdl:userName>
                    <jsdl:password>${agent:password.mssqluser}</jsdl:password>[1]
                </jsdldatabase:credentials>
            </jsdldatabase:sqlActionInfo>
        </jsdldatabase:database>
  </jsdl:application>
</jsdl:jobDefinition>
DESCRIPTION "Definida usando o composer."
RECOVERY STOP
```

Nota: (1) A senha para o usuário `mssqluser` foi definida com o comando do utilitário `param` no arquivo de variáveis no agente que executará a tarefa. Ela será resolvida no tempo de execução com o valor definido.

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

o Guia do Usuário do Console do Dynamic Workload, seção Criando Definições de Tarefa.

Para obter informações adicionais sobre como criar e editar objetos de planejamento, consulte:

o Guia do Usuário do Console do Dynamic Workload, seção Projetando sua Carga de Trabalho.

Definição de Tarefa - Tarefas Java

Esta seção descreve os atributos necessários e opcionais para tarefas Java. Cada definição de tarefa tem o seguinte formato e argumentos:

Tabela 37. Atributos Obrigatórios e Opcionais para a Definição de uma Tarefa Java.

Atributo	Descrição/valor	Exigido
nome do aplicativo	java	✓
jarPath	O diretório no qual os arquivos jar estão armazenados. Isso inclui todos os arquivos jar armazenados no diretório especificado e todos os subdiretórios.	
className	O nome da classe que a tarefa deve executar.	✓
chave de parâmetro	Os parâmetros a serem usados ao executar a classe Java.	

Para obter informações adicionais sobre como desenvolver uma tarefa Java, consulte *Tivoli Workload Automation: Developer's Guide: Extending Tivoli Workload Automation*.

O exemplo a seguir mostra uma tarefa que executa uma classe com nome `com.ibm.test.Test` e parâmetro `failExecution`:

```
$JOBS
AGENT#JAVA
TASK
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdljava="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdljava" name="java">
  <jsdl:application name="java">
    <jsdljava:java>
            <jsdljava:javaParms>
                <jsdljava:jarPath>C:\JavaExecutors</jsdljava:jarPath>
                <jsdljava:className>com.ibm.test.Test</jsdljava:className>
                <jsdljava:parameters>
                    <jsdljava:parameter key="input">failExecution</jsdljava:parameter>
                </jsdljava:parameters>
            </jsdljava:javaParms>
       </jsdljava:java>
  </jsdl:application>
</jsdl:jobDefinition> DESCRIPTION "Definida usando o composer."
 RECOVERY STOP
```

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

o Guia do Usuário do Console do Dynamic Workload, seção Criando Definições de Tarefa.

Para obter informações adicionais sobre como criar e editar objetos de planejamento, consulte:

o Guia do Usuário do Console do Dynamic Workload, seção Projetando sua Carga de Trabalho.

Definição de Tarefa - Tarefas Executáveis

Esta seção descreve os atributos necessários e opcionais para tarefas executáveis. Cada definição de tarefa tem o seguinte formato e argumentos:

Tabela 38. Atributos Obrigatórios e Opcionais para a Definição de uma Tarefa Executável.

Atributo	Descrição/valor	Exigido
nome do aplicativo	executável	✓
interactive	Especifique se a tarefa requer intervenção do usuário. Essa opção se aplica somente a tarefas que são executadas nos sistemas operacionais Windows.	✓
valor	Especifique o nome e o valor de um ou mais argumentos.	
script	Digite um script para ser executado pela tarefa. O script é criado e executado quando tarefa é executada. É possível especificar os argumentos nessa tag ou é possível digitá-los na tag **value** e chamá-los no script.	✓
sufixo	Especifique a extensão de nome de arquivo para o script a ser executado pela tarefa. Essa opção se aplica somente a tarefas que são executadas nos sistemas operacionais Windows. Não insira o "." no início do nome da extensão.	

O exemplo a seguir mostra uma tarefa que efetua ping em dois websites. O endereço dos websites é definido na tag **value** e chamado na tag **script**. Essa tarefa tem um relacionamento de afinidade com a tarefa `affine_test`, o que significa que essa tarefa é executada na mesma estação de trabalho que o `affine_test`:

```
$JOBS
AGENT#EXECUTABLE
TASK
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle" name="executable">
  <jsdl:application name="executable">
    <jsdle:executable interactive="false" workingDirectory="c:\">
            <jsdle:arguments>
                <jsdle:value>www.mysite.com</jsdle:value>
                <jsdle:value>www.yoursite.com</jsdle:value>
            </jsdle:arguments>
            <jsdle:script>ping %1 ping %2</jsdle:script>
        </jsdle:executable>
  </jsdl:application>
</jsdl:jobDefinition>
DESCRIPTION "Definida usando o composer."
TWSAFFINITY "affine_test"
RECOVERY STOP
```

O exemplo a seguir mostra uma tarefa que executa um script `vbs` em sistemas operacionais Windows. A extensão do nome do arquivo é definida no atributo `sufixo` da tag `script`:

```
WIN_WKS1#VBS_NAT1
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle" name="executable">
  <jsdl:application name="executable">
    <jsdle:executable interactive="true" workingDirectory="c:\tws">
            <jsdle:script suffix="vbs">Wscript.Echo "ciao"</jsdle:script>
        </jsdle:executable>
  </jsdl:application>
</jsdl:jobDefinition>
 RECOVERY STOP
```

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

o Guia do Usuário do Console do Dynamic Workload, seção Criando Definições de Tarefa.

Para obter informações adicionais sobre como criar e editar objetos de planejamento, consulte:

o Guia do Usuário do Console do Dynamic Workload, seção Projetando sua Carga de Trabalho.

Definição de Tarefa - Tarefas de comando remoto

Para ler um cenário da vida real comum que atinge metas de negócios, incluindo a implementação de uma tarefa de comando remoto, consulte http://pic.dhe.ibm.com/infocenter/tivihelp/v47r1/index.jsp?topic=/com.ibm.tivoli.itws.doc_8.6.0.2/scen/awsbsscp_rc.html.

Esta seção descreve os atributos necessários e opcionais para tarefas de comando remoto. Cada definição de tarefa tem o seguinte formato e argumentos:

Tabela 39. Atributos necessários e opcionais para a definição de uma tarefa de comando remoto

Atributo	Descrição/valor	Exigido
nome do aplicativo	remotecommand	✓

userName	O nome de usuário autorizado a iniciar uma conexão no computador remoto usando o protocolo definido. Como uma alternativa aos valores reais de codificação permanente, é possível parametrizar de uma das seguintes maneiras:
	• Insira um *nome de usuário* especificado no banco de dados com a definição de usuário (aplicável a todos os sistemas operacionais nesse tipo de tarefa) e insira a instrução:
	`<jsdl:password>${password:username}</jsdl:password>`
	A senha é recuperada da definição de usuário do *nome de usuário* no banco de dados e resolvida em tempo de execução. Consulte "Utilizando Definições do Usuário sobre Tipos de Tarefas com Opções Avançadas" na página 195 para obter mais detalhes.
	Também é possível especificar o usuário de uma estação de trabalho diferente e usar a seguinte sintaxe para a senha:
	`<jsdl:password>${password:workstation#username}</jsdl:password>`
	• Insira um usuário e senha definidos com o comando do utilitário `param` localmente no agente dinâmico que executará a tarefa (se a tarefa for enviada para um conjunto ou para um conjunto dinâmico, a definição deve estar presente em todos os agentes do conjunto). Desde que você tenha definido o nome do usuário com a variável *user* e uma senha, as instruções da credencial correspondente seriam:
	`<jsdl:userName>${agent:user}</jsdl:userName>`
	`<jsdl:password>${agent:password.user}</jsdl:password>`
		✓

password	A senha do usuário autorizado. A senha é criptografada quando a tarefa é criada. Consulte a descrição para userName para obter detalhes adicionais.	
nome do servidor	O nome do host do computador no qual a instância de comando remoto está em execução.	✓
porta	O número da porta do computador remoto no qual o comando é executado.	✓

Tabela 39. Atributos necessários e opcionais para a definição de uma tarefa de comando remoto (continuação)

Atributo	Descrição/valor	Exigido
protocol	Valores possíveis: **AUTO** O protocolo é selecionado automaticamente entre os protocolos existentes: SSH, Windows, RSH e REXEC. O produto tenta usar o protocolo SSH primeiro. Se esse protocolo falhar, o protocolo do Windows será usado. Ao usar o SSH, o caminho deve estar no formato SSH. Neste caso, o servidor ssh Cygwin está montado em /home/Administrator. **SSH** Um protocolo de rede que fornece funções de acesso a arquivos, de transferência de arquivos e de gerenciamento de arquivos em qualquer fluxo de dados. **WINDOWS** O protocolo de compartilhamento de arquivo da Microsoft. Use a sintaxe samba para especificar o caminho. Compartilhe a pasta que contém os arquivos que você deseja transferir. **RSH** Remote Shell Protocol (rsh) é um protocolo que permite que um usuário execute comandos em um sistema remoto sem precisar efetuar login no sistema. **REXEC** O servidor Remote Execution (REXEC) é um aplicativo Protocolo de Controle de Transmissões/Protocolo da Internet (TCP/IP) que permite que um usuário cliente envie comandos do sistema para um sistema remoto. O Protocolo Remote Execution (REXEC) permite o processamento desses comandos ou programas em qualquer host na rede. O host local então recebe os resultados do processamento de comando.	
Caminho de arquivo keystore	O caminho completo do arquivo keystore que contém a chave privada usada para fazer a conexão. Um keystore é um banco de dados de chaves. As chaves privadas em um keystore possuem uma cadeia de certificados associada a elas que autentica a chave pública correspondente no servidor remoto. Um keystore também contém certificados de entidades confiáveis. Aplicável apenas ao protocolo SSH.	
senha do keystore	A senha que protege a chave privada e é necessária para fazer a conexão. Este atributo é necessário apenas se você especificar um caminho de arquivo keystore. Se a combinação de caminho de arquivo keystore e senha do keystore falhar ao fazer uma conexão, será feita uma tentativa usando o nome de usuário e senha que correspondem ao usuário autorizado a iniciar uma conexão no computador remoto.	✓
comando	Digite o comando a ser enviado no computador remoto. .	✓
environment	Os arquivos de saída padrão e de erro padrão para o comando remoto. Estes arquivos estão localizados no agente, não localmente nas estações de trabalho nas quais o comando remoto é executado. Certifique-se de que tenha direitos de gravação nos diretórios especificados, caso contrário, nenhum arquivo será criado. **Saída Padrão** Especifique o caminho e o nome do arquivo em que a saída padrão para o comando deve ser salva. Especifique um nome absoluto do caminho ou um nome do caminho relativo ao diretório ativo. O arquivo é sobrescrito sempre que o comando produz uma nova saída. **Erro Padrão** Especifique o caminho e nome do arquivo em que o erro padrão para o comando deve ser salvo. Especifique um nome absoluto do caminho ou um nome do caminho relativo ao diretório ativo. O arquivo é sobrescrito sempre que o comando produz um novo erro.	

O exemplo a seguir mostra a seção "aplicativo" JSDL de uma definição de tarefa de amostra para uma tarefa do comando remoto:

```
$JOBS
NC112016#REMCMD
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
   xmlns:jsdlremotecommand="http://www.ibm.com/xmlns/prod/scheduling/1.0/
    jsdlremotecommand" name="REMOTECOMMAND">
  <jsdl:application name="remotecommand">
    <jsdlremotecommand:remotecommand>
   <jsdlremotecommand:RemoteCommandParameters>
    <jsdlremotecommand:taskPanel>
     <jsdlremotecommand:command>ping -c 10 localhost </jsdlremotecommand:command>
    </jsdlremotecommand:taskPanel>
    <jsdlremotecommand:environmentPanel>
     <jsdlremotecommand:standardOutput>stdout</jsdlremotecommand:standardOutput>
     <jsdlremotecommand:standardError>stderr</jsdlremotecommand:standardError>
    </jsdlremotecommand:environmentPanel>
    <jsdlremotecommand:serverPanel>
     <jsdlremotecommand:serverInfo>
      <jsdlremotecommand:serverName>9.168.112.16</jsdlremotecommand:serverName>
      <jsdlremotecommand:port>23</jsdlremotecommand:port>
      <jsdlremotecommand:protocol>ssh</jsdlremotecommand:protocol>
     </jsdlremotecommand:serverInfo>
     <jsdlremotecommand:credentials>
      <jsdl:userName>userName</jsdl:userName>
      <jsdl:password>{aes}mv0GJqOHWo8lbuhcpFaluL9RkGQKrYvTiAUpKTMgp90=
            </jsdl:password>
     </jsdlremotecommand:credentials>
     <jsdlremotecommand:certificates>
```

```
      <jsdlremotecommand:keystoreFilePath>/var/keyStoreFile</jsdlremotecommand:
                keystoreFilePath>
      <jsdlremotecommand:keystorePassword>pwd</jsdlremotecommand:keystorePassword>
     </jsdlremotecommand:certificates>
    </jsdlremotecommand:serverPanel>
   </jsdlremotecommand:RemoteCommandParameters>
  </jsdlremotecommand:remotecommand>
  </jsdl:application>
</jsdl:jobDefinition>
 RECOVERY STOP
```

Você planeja tarefas de comando remoto do Tivoli Workload Scheduler definindo-as em fluxos de tarefas. Inclua a tarefa em um fluxo de tarefas com todos os argumentos de planejamento necessários e envie-a. As tarefas podem ser enviadas em um ambiente distribuído usando o Dynamic Workload Console ou a linha de comandos **conman**. As tarefas podem ser enviadas em um ambiente z/OS usando o Dynamic Workload Console. Após o envio, quando a tarefa estiver em execução (status **EXEC**), não será possível executar **kill** na tarefa. A ação **kill** é ignorada por **conman** e a tarefa continua em execução.

Em ambientes distribuídos, se o agente Tivoli Workload Scheduler se tornar indisponível durante o envio da tarefa de comando remoto do Tivoli Workload Scheduler ou durante a execução da tarefa, o Tivoli Workload Scheduler designará o status Error ou ABEND à tarefa quando o agente for reiniciado.

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

o Guia do Usuário do Console do Dynamic Workload, seção Criando Definições de Tarefa.

Para obter informações adicionais sobre como criar e editar objetos de planejamento, consulte:

o Guia do Usuário do Console do Dynamic Workload, seção Projetando sua Carga de Trabalho.

Definição de Tarefa - Tarefas de Método de Acesso

Esta seção descreve os atributos necessários e opcionais para tarefas de método de acesso. Cada definição de tarefa tem o seguinte formato e argumentos:

Tabela 40. Atributos Obrigatórios e Opcionais para a Definição de uma Tarefa de Método de Acesso

Atributo	Descrição/valor	Exigido
nome do aplicativo	xajob	✓
accessMethod	O nome do método de acesso usado para comunicar-se com o sistema externo para iniciar a tarefa e retornar o status da tarefa.	✓
target	O nome de um arquivo de opções.	
taskString	Comando a ser interpretado pelo método selecionado. The maximum line length is 8 KB.	✓

Tabela 40. Atributos Obrigatórios e Opcionais para a Definição de uma Tarefa de Método de Acesso (continuação)

Atributo	Descrição/valor	Exigido

credenciais	O nome e a senha do usuário que está executando essa tarefa. Como uma alternativa aos valores reais de codificação permanente, é possível parametrizar de uma das seguintes maneiras:
	• Insira um *nome de usuário* especificado no banco de dados com a definição de usuário (aplicável a todos os sistemas operacionais nesse tipo de tarefa) e insira a instrução:
	`<jsdl:password>${password:username}</jsdl:password>`
	A senha é recuperada da definição de usuário do *nome de usuário* no banco de dados e resolvida em tempo de execução. Para obter detalhes adicionais, consulte "Utilizando Definições do Usuário sobre Tipos de Tarefas com Opções Avançadas" na página 195.
	Também é possível especificar o usuário de uma estação de trabalho diferente e usar a seguinte sintaxe para a senha:
	`<jsdl:password>${password:workstation#username}</jsdl:password>`
	• Insira um usuário e senha definidos com o comando do utilitário `param` localmente no agente dinâmico que executará a tarefa (se a tarefa for enviada para um conjunto ou para um conjunto dinâmico, a definição deve estar presente em todos os agentes do conjunto). Se tiver definido o nome de usuário com a variável *user* e uma senha, as instruções da credencial correspondente são:
	`<jsdl:userName>${agent:user}</jsdl:userName>`
	`<jsdl:password>${agent:password.user}</jsdl:password>`
	As variáveis usuário e senha são resolvidas no agente no tempo de execução. Para obter detalhes adicionais, consulte "Definindo Variáveis e Senhas para Resolução Local em Agentes Dinâmicos" na página 477.	

O exemplo a seguir mostra uma tarefa que cria um arquivo na pasta /methods usando uma tarefa de método de acesso padrão:

```
$JOBS
AGENT#XA_JOB
TASK
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition
xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdlxa=
"http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdlxa"
name="xajob">
  <jsdl:application name="xajob">
    <jsdlxa:xajob accessMethod="unixlocl" target="optionFile">
       <jsdlxa:taskString>touch file</jsdlxa:taskString>
       <jsdlxa:credentials>
         <jsdlxa:userName>TestUser</jsdlxa:userName>
         <jsdlxa:password>{aes}IEr/DES8wRzQEij1ySQBfUR587QBxM0iwfQ1EWJaDds=</jsdlxa:password>
       </jsdlxa:credentials>
     </jsdlxa:xajob>
  </jsdl:application>
</jsdl:jobDefinition>
DESCRIPTION "Definida usando o composer."
RECOVERY STOP
```

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

o Guia do Usuário do Console do Dynamic Workload, seção Criando Definições de Tarefa.

Para obter informações adicionais sobre como criar e editar objetos de planejamento, consulte:

o Guia do Usuário do Console do Dynamic Workload, seção Projetando sua Carga de Trabalho.

Definição de Tarefa - Tarefas z/OS

Esta seção descreve os atributos necessários e opcionais para tarefas z/OS. Uma tarefa do z/OS executa o comando especificado na guia JCL em um sistema JCL. Esse tipo de tarefa é executada apenas no Tivoli Workload Scheduler distribuído - Agent for z/OS. Cada definição de tarefa tem o seguinte formato e argumentos:

Tabela 41. Os atributos necessários e opcionais para a definição de uma tarefa z/OS.

Atributo	Descrição/valor	Exigido
nome do aplicativo	jcl	✓
byDefinition	O tipo de envio de tarefa. Esta é o único tipo de envio suportado.	
jclDefinition	A operação a ser executada no sistema JCL.	✓

O exemplo a seguir mostra uma tarefa que retorna o status do sistema JCL:

```
ZOSAGENT#JCLDEF
 TASK
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition
xmlns:jsdl=="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
 xmlnss:jsdljcl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdljcl">
<jsdl:application name="jcl">
<jsdljcl:jcl>
<jsdljcl:JCLParameters>
<jsdljcl:jcl>
<jsdljcl:byRefOrByDef>
<jsdljcl:byDefinition>
<jsdljcl:jclDefinition>//NORMAL JOB ,'TWS
JOB',CLASS=A,MSGCLASS=A,>
// MSGLEVEL=(1,1)
//*
//STEP1 EXEC PGM=IEFBR14</jsdljcl:jclDefinition>
</jsdljcl:byDefinition>
</jsdljcl:byRefOrByDef>
</jsdljcl:jcl>
</jsdljcl:JCLParameters>
<jsdljcl:JOBParameters>
<jsdljcl:jobStreamName>${tws.jobstream.name}jsdljcl:jobStreamName>${tws.jobstream.name}>
<jsdljcl:inputArrival>${tws.job.ia}jsdljcl:inputArrival>${tws.job.ia}>
</jsdljcl:JOBParameters>
</jsdljcl:jcl>
</jsdl:application>
</jsdl:jobDefinition>
 DESCRIPTION "Sample JCL Job Definition"
```

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

o Guia do Usuário do Console do Dynamic Workload, seção Criando Definições de Tarefa.

Para obter informações adicionais sobre como criar e editar objetos de planejamento, consulte:

o Guia do Usuário do Console do Dynamic Workload, seção Projetando sua Carga de Trabalho.

Definição de Tarefa - Tarefas do IBM i

Esta seção descreve os atributos necessários e opcionais para tarefas do IBM i. Uma tarefa IBM i executa o comando especificado na guia IBM i em um sistema IBM i (anteriormente conhecido como sistema operacional AS/400 e i5). Cada definição de tarefa tem o seguinte formato e argumentos:

Tabela 42. Atributos Obrigatórios e Opcionais para a Definição de uma Tarefa do IBM i.

Atributo	Descrição/valor	Exigido
nome do aplicativo	ibmi	✓

Tabela 42. Atributos Obrigatórios e Opcionais para a Definição de uma Tarefa do IBM i. (continuação)

Atributo	Descrição/valor	Exigido
comando	O comando a ser executado no sistema IBM i.	✓

O exemplo a seguir mostra uma tarefa que retorna o status do sistema IBM i:

```
$JOBS
AGENT#IBM_I
TASK
   <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
 xmlns:jsdlibmi="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdlibmi"
name="ibmi">
  <jsdl:application name="ibmi">
    <jsdlibmi:ibmi>
            <jsdlibmi:IBMIParameters>
                <jsdlibmi:Task>
                    <jsdlibmi:command>wrksyssts</jsdlibmi:command>
                </jsdlibmi:Task>
            </jsdlibmi:IBMIParameters>
        </jsdlibmi:ibmi>
  </jsdl:application>
</jsdl:jobDefinition>
RECOVERY STOP
```

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

o Guia do Usuário do Console do Dynamic Workload, seção Criando Definições de Tarefa.

Para obter informações adicionais sobre como criar e editar objetos de planejamento, consulte:

o Guia do Usuário do Console do Dynamic Workload, seção Projetando sua Carga de Trabalho.

Definição de Tarefa - OSLC Automation

Use a tarefa do OSLC Automation para chamar qualquer provedor OSLC que esteja implementando o OSLC Automation Specification. Para obter informações detalhadas, consulte http://open-services.net/wiki/automation/OSLC-Automation-Specification-Version-2.0/.

Antes de definir definições de tarefa do OSLC Automation, você deve executar algumas etapas de pré-requisito, conforme explicado em "Etapas de Pré-requisito para Criar Tarefas do OSLC Automation e do OSLC Provisioning no *Dynamic Workload Console User's Guide*.

Esta seção descreve os atributos necessários e opcionais para tarefas do OSLC Automation. Cada definição de tarefa tem o seguinte formato e argumentos:

Tabela 43. Atributos necessários e opcionais para a definição de uma tarefa do OSLC Automation.

Atributo	Descrição/valor	Exigido
URI do Registry Services	O endereço do Registry Services (por exemplo, `https://myhost.mydomain:16311/oslc/pr`).	
Nome de Usuário do Registry Services	O usuário que está se conectando ao Registry Services.	
Senha do Registry Services	A senha associada ao usuário que está se conectando ao Registry Services.	
URI do Service Provider	O endereço do Service Provider.	✓

Tabela 43. Atributos necessários e opcionais para a definição de uma tarefa do OSLC Automation. (continuação)

Atributo	Descrição/valor	Exigido
Nome de Usuário do Service Provider	O usuário que está se conectando ao Service Provider.	
Senha do Provedor de Serviços	A senha associada ao usuário que está se conectando ao Service Provider.	
Solicitação	A representação de RDF da solicitação de automação.	✓

O exemplo a seguir mostra uma tarefa que planeja um fluxo de tarefas do Tivoli Workload Scheduler:

```
$JOBS
WKS#AUTOMATION
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
     xmlns:jsdloslcautomation="
http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdloslcautomation" name="OSLCAUTOMATION">
  <jsdl:application name="oslcautomation">
    <jsdloslcautomation:oslcautomation>
              <jsdloslcautomation:OSLCAutomationParameters>
                    <jsdloslcautomation:AutomationRequest>
                          <jsdloslcautomation:automationRequestGroup>
                                <jsdloslcautomation:automationRequestBody>

<!-- add the rdf representation of the resource -->

</jsdloslcautomation:automationRequestBody>
                               </jsdloslcautomation:automationRequestGroup>
                        </jsdloslcautomation:AutomationRequest>
                        <jsdloslcautomation:ConnectionInfo>
                            <jsdloslcautomation:ServiceProviderCatalogGroup>
                                  <jsdloslcautomation:catalogURI>
                                   https://registryserviceshost.domain:16311/oslc/pr>
                                  </jsdloslcautomation:catalogURI>
                                  <jsdloslcautomation:username>registryUser</jsdloslcautomation:
                                     username>
                                  <jsdloslcautomation:password>registryPassword</jsdloslcautomation:
                                     password>
                               </jsdloslcautomation:ServiceProviderCatalogGroup>
                               <jsdloslcautomation:ServiceProviderGroup>
                                     <jsdloslcautomation:serviceProviderURI>
https://serviceprovideraddress.domain:16310/oslc/providers/1360665198982</jsdloslcautomation:
                                      serviceProviderURI>
                                    <jsdloslcautomation:usernameSP>
                                     serviceProviderUser
                                    </jsdloslcautomation:usernameSP>
                                    <jsdloslcautomation:passwordSP>
                                     serviceProviderPassword
                                    </jsdloslcautomation:passwordSP>
                               </jsdloslcautomation:ServiceProviderGroup>
                          </jsdloslcautomation:ConnectionInfo>
              </jsdloslcautomation:OSLCAutomationParameters>
           </jsdloslcautomation:oslcautomation>
  </jsdl:application>
</jsdl:jobDefinition>
```

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

o Guia do Usuário do Console do Dynamic Workload, seção Criando Definições de Tarefa.

Para obter informações adicionais sobre como criar e editar objetos de planejamento, consulte:

o Guia do Usuário do Console do Dynamic Workload, seção Projetando sua Carga de Trabalho.

Definição de Tarefa - OSLC Provisioning

Use a tarefa do OSLC Provisioning para chamar qualquer provedor OSLC que esteja implementando o OSLC Provisioning Specification.

Antes de definir definições de tarefa do OSLC Provisioning, você deve executar algumas tarefas de pré-requisito, conforme explicado em "Etapas de Pré-requisito para Criar Tarefas do OSLC Automation e do OSLC Provisioning no *Dynamic Workload Console User's Guide*.

Esta seção descreve os atributos necessários e opcionais para tarefas do OSLC Provisioning. Cada definição de tarefa tem o seguinte formato e argumentos:

Tabela 44. Atributos necessários e opcionais para a definição de uma tarefa do OSLC Provisioning.

Atributo	Descrição/valor	Exigido
Registry Services URI	O endereço do Registry Services (por exemplo, `https://myhost.mydomain:16311/odlc/pr`).	✓
Registry ServicesNome do usuário	O usuário que está se conectando ao Registry Services.	✓
Senha do Registry Services	A senha associada ao usuário que está se conectando ao Registry Services.	✓
URI do Service Provider	O endereço do Service Provider.	✓
Nome de Usuário do Service Provider	O usuário que está se conectando ao Service Provider.	✓
Senha do Provedor de Serviços	A senha associada ao usuário que está se conectando ao Service Provider.	✓
Instância	A representação RDF da instância a ser implementada.	✓

O exemplo a seguir mostra uma tarefa que planeja o fornecimento de um padrão do sistema:

```
WKS#PROVSAMPLETASK
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
 xmlns:jsdloslcprovisioning="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdloslcprovisioning"
 name="OSLCPROVISIONING">
 <jsdl:application name="oslcprovisioning">
   <jsdloslcprovisioning:oslcprovisioning>
    <jsdloslcprovisioning:OSLCProvisioningParameters>
     <jsdloslcprovisioning:actionPanel>
      <jsdloslcprovisioning:instanceFromTemplate>
       <jsdloslcprovisioning:instance>
       <!-- RDF definition of the instance>
&lt;?ml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?>
&lt;rf:RDF
     xmlns:rdf=&quot;http://www.w3.org/1999/02/22-rdf-syntax-ns#&quot;
     xmlns:oslc=&quot;http://open-services.net/ns/core#&quot;
     xmlns:sco=&quot;http://jazz.net/ns/ism/provisioning/sco#&quot;
     xmlns:oslc_auto=&quot;http://open-services.net/ns/auto#&quot;
     xmlns:dcterms=&quot;http://purl.org/dc/terms/&quot;
     xmlns:rdfs=&quot;http://www.w3.org/2000/01/rdf-schema#&quot; >
    &lt;rf:Description rdf:about=&quot;
http://myServiceProvider.domain:31115/CLIModelWeb/OSLC/BatchApplicationInstance/BatchApplication/
  0f01af24-72e0-3c4b-b95c-18f908c76898&quot;>
     &lt;olc_auto:parameterDefinition rdf:nodeID=&quot;A1&quot;/>
     &lt;olc_auto:parameterDefinition rdf:nodeID=&quot;A2&quot;/>
     &lt;rf:type rdf:resource=&quot;http://jazz.net/ns/ism/provisioning/sco#Entity&quot;/>
     &lt;olc_auto:parameterDefinition rdf:nodeID=&quot;A4"/>
     <dcterms:identifier>0f01af24-72e0-3c4b-b95c-18f908c76898</dcterms:identifier>
     <oslc_auto:parameterDefinition rdf:nodeID="A5"/>
     <dcterms:title>InstanceName</dcterms:title>
     <oslc_auto:parameterDefinition rdf:nodeID="A0"/>
   </rdf:Description>
   <rdf:Description rdf:nodeID="A5">
     <oslc:name>XML</oslc:name>
     <oslc:value>&lt;?xml version=&quot1.0&quot encoding=&quotUTF-8&quot;?&gt;
        &lt;model:TWSBatchApplicationInstance xmlns:model=&quot;
http://www.ibm.com/xmlns/prod/scheduling/1.0/Model&quot&gt;
        &lt;model:Name&gt;InstanceName&lt;/model:Name&gt;
             <!--Aqui se inicia a definição de tarefas, os fluxos de tarefas, etc. contidos nesta instância>
             ...
             <!--Aqui se inicia a definição >
 &lt;/model:TWSBatchApplicationInstance&gt;&lt;/slc:value>
```

```
  &lt;olc:defaultValue&gt;&lt;/slc:defaultValue>
     &lt;rdf:type rdf:resource=&quot;http://open-services.net/ns/core#Property&quot;/>
   </rdf:Description>
   <!--Mapeamento da instância>
    &lt;rdf:Description rdf:nodeID=&quot;A0&quot;>
    &lt;oslc:name>JOB_SAMPLE&lt;/oslc:name>
        &lt;oslc:value>JOB_TARGET&lt;/oslc:value>
    &lt;oslc:defaultValue>JOB_TARGET&lt;/oslc:defaultValue>
    &lt;rdf:type rdf:resource=&quot;http://open-services.net/ns/core#Property&quot;/>
  &lt;/rdf:Description>
  <!--Aqui continua a definição de tarefas, fluxos de tarefas, etc. contidos nesta instância>
   ...
  <!--Aqui termina o mapeamento>
   &lt;rdf:Description rdf:nodeID=&quot;A7&quot;>
     &lt;sco:node rdf:resource=&quot;
http://thinklinux:31115/CLIModelWeb/OSLC/BatchApplicationInstance/BatchApplication/
      0f01af24-72e0-3c4b-b95c-18f908c76898&quot;/>
     &lt;rdf:type rdf:resource=&quot;http://jazz.net/ns/ism/provisioning/sco#Template&quot;/>
    &lt;/rdf:Description>
   &lt;rdf:Description rdf:nodeID=&quot;A1&quot;>
     &lt;oslc:name>ICON&lt;/oslc:name>
     &lt;oslc:value>../js/images/default.png&lt;/oslc:value>
     &lt;oslc:defaultValue>../js/images/default.png&lt;/oslc:defaultValue>
    &lt;rdf:type rdf:resource=&quot;http://open-services.net/ns/core#Property&quot;/>
   &lt;/rdf:Description> &lt;/rdf:RDF></jsdloslcprovisioning:instance>
      </jsdloslcprovisioning:instanceFromTemplate>
     </jsdloslcprovisioning:actionPanel>
     <jsdloslcprovisioning:connectionInfo>
      <jsdloslcprovisioning:serviceProviderCatalog>
       <jsdloslcprovisioning:catalogURI>
            https://myregistry.domain:16311/oslc/pr</jsdloslcprovisioning:catalogURI>

       <jsdloslcprovisioning:username>registryServicesUser</jsdloslcprovisioning:username>

      <jsdloslcprovisioning:password>registryServicesPassword</jsdloslcprovisioning:password>
                                           </jsdloslcprovisioning:serviceProviderCatalog>
                                            <jsdloslcprovisioning:serviceProvider>
                                            <jsdloslcprovisioning:serviceProviderURI>
https://myregistry.domain:16311/oslc/providers/1380617052297
                                                 </jsdloslcprovisioning:serviceProviderURI>

       <jsdloslcprovisioning:usernameSP>myServiceProviderUser</jsdloslcprovisioning:usernameSP>
       <jsdloslcprovisioning:passwordSP>myServiceProviderPassword</jsdloslcprovisioning:passwordSP>
                                             </jsdloslcprovisioning:serviceProvider>
                                     </jsdloslcprovisioning:connectionInfo>
                                </jsdloslcprovisioning:OSLCProvisioningParameters>
                           </jsdloslcprovisioning:oslcprovisioning>
  </jsdl:application>
 </jsdl:jobDefinition>
```

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

o Guia do Usuário do Console do Dynamic Workload, seção Criando Definições de Tarefa.

Para obter informações adicionais sobre como criar e editar objetos de planejamento, consulte:

o Guia do Usuário do Console do Dynamic Workload, seção Projetando sua Carga de Trabalho.

Utilizando Variáveis e Parâmetros nas Definições de Tarefa

Uma variável é um objeto de planejamento que faz parte de uma tabela de variáveis e é definido no banco de dados Tivoli Workload Scheduler. Ela pode ser utilizada por todos os agentes no domínio contanto que os usuários tenham autorização apropriada no arquivo de segurança.

Um parâmetro é definido e utilizado localmente em um agente (com o comando do utilitário `parms`).

Variáveis e parâmetros possuem os seguintes usos e limitações nas definições de tarefa:

- Variáveis e parâmetros têm permissão nos valores para as palavras-chave **streamlogon**, **scriptname**, **docommand** e **abendprompt**.
- Uma variável ou parâmetro pode ser utilizado como uma cadeia inteira ou como parte dela.
- Várias variáveis e parâmetros têm permissão em um único campo.
- Coloque os nomes de variáveis entre sinais de intercalação (^) e coloque a cadeia inteira entre aspas. Assegure-se de que os caracteres de acento circunflexo não sejam precedidos por uma barra invertida na cadeia. Se necessário, inclua a barra invertida na definição da variável ou parâmetro.
- Coloque os nomes de parâmetros entre aspas simples (') no UNIX e coloque a cadeia inteira entre aspas.
- Consulte "Definição de Variável e Parâmetro" na página 198 para obter informações adicionais e exemplos.

No exemplo a seguir, uma variável denominada **mis** é utilizada no valor **streamlogon**:

```
$jobs
cpu1#bkup
     scriptname "/usr/mis/scripts/bkup"
     streamlogon "^mis^"
     recovery continue after recjob1
```

Para obter outros exemplos, consulte "Definição de Variável e Parâmetro" na página 198.

Definição do Usuário

Os nomes de usuário que são utilizados como o valor **streamlogon** para as definições de tarefa do Windows devem ter definições do usuário. Isso não é necessário para usuários que executem tarefas em outros sistemas operacionais. Se você estiver usando o tipos de tarefas com opções avançadas, é possível usar esse valores independentemente do sistema operacional. Para obter informações adicionais, consulte a "Utilizando Definições do Usuário sobre Tipos de Tarefas com Opções Avançadas" na página 195.

Nota: Se você tiver a opção global `enAddUser` configurada como `"yes"`, a definição de usuário será automaticamente incluída no plano após você criar ou modificar a definição de usuário no banco de dados.

Sintaxe

Cada definição de usuário tem o seguinte formato e argumentos:

```
username[workstation#][domain\]username[@internet_domain]
password "password"
end
```

Argumentos

username

[*workstation***#]***username*

estação de trabalho
Especifica a estação de trabalho na qual o usuário tem

permissão para lançar jobs. O sinal de número é necessário. O padrão é em branco, significando todas as estações de trabalho.

username
Especifica o nome do usuário do Windows. O valor do campo *username* pode conter até 47 caracteres.

[*workstation***#]***domain**username*

estação de trabalho
Especifica a estação de trabalho na qual o usuário tem permissão para lançar jobs. O sinal de número é necessário. O padrão é em branco, significando todas as estações de trabalho.

*domain**username*

Especifica o nome do usuário do domínio do Windows. O valor do campo*domain**username* pode conter até 47 caracteres.

[*workstation***#]***username***@***internet_domain*

estação de trabalho
Especifica a estação de trabalho na qual o usuário tem permissão para lançar jobs. O sinal de número é necessário. O padrão é em branco, significando todas as estações de trabalho.

*username***@***internet_domain*

Especifica o nome do usuário no formato User Principal Name (UPN). O formato UPN é o nome de um usuário do sistema em um formato de endereço de email. O nome de usuário é seguido pelo símbolo "@" e pelo nome do domínio da Internet ao qual o usuário está associado. O valor do campo *username***@***internet_domain* pode conter até 47 caracteres.

Nota:

Se você definir um usuário para sistemas operacionais **Windows**:

- Os nomes de usuário fazem distinção entre maiúsculas e minúsculas. Além disso, o usuário deve estar autorizado a efetuar logon na estação de trabalho na qual o Tivoli Workload Scheduler ativa tarefas, e ter a permissão para **Efetuar logon como lote**.
- Se o nome de usuário não for exclusivo, assume-se como um usuário local, um usuário do domínio ou um usuário do domínio confiável, nessa ordem.

password
Especifica a senha do usuário. A senha pode conter até 31 caracteres e deve ser colocada entre aspas duplas. Para indicar uma senha nula, utilize duas aspas duplas consecutivas sem espaços em branco entre elas, **""**. Quando uma definição do usuário tiver sido salva, não será possível ler a senha. Usuários com privilégios de segurança apropriados podem modificar ou excluir um usuário, mas as informações de senha nunca são exibidas.

Exemplos

O exemplo a seguir define quatro usuários:

```
username joe
     password "okidoki"
end
#
username server#jane
     password "okitay"
end
#
username dom1\jane
     password "righto"
end
#
username jack
     password ""
end
#
username administrator@twsbvt.com
     password "internetpwd"
end
#
username serverA#dom1\jack
     password "righto"
end
#
username serverB#user1@twsbvt.com
     password "internetpwd"
end
#
```

Comentários

Senhas extraídas com o comando `composer extract` são de uso limitado. Quando você executa o comando `composer extract` em uma definição de usuário, a senha é ofuscada com a palavra-chave reservada "**********". Se você tentar executar os comandos do composer `import`, `replace` ou `modify` sobre uma senha de usuário extraída, a substituição de senha não terá efeito e a senha antiga será mantida. Além disso, se você tentar executar os comandos do composer `create`, `new` ou `add` sobre um usuário em que a senha é igual à palavra-chave reservada "**********", o erro a seguir será retornado:

```
AWSJCL521E A senha especificada para o usuário do Windows "USER_NAME" não
está em conformidade com os requisitos da política de segurança de senha.
```

Note que a palavra-chave reservada é uma sequência de dez asteriscos (*). Não é possível inserir uma sequência de dez asteriscos como senha, mas é possível ter uma senha com qualquer outro número de asteriscos.

Para corrigir esse problema, certifique-se de executar o`composer extract` com a opção ;password.

Consulte também

Para obter mais informações sobre como executar a mesma tarefa a partir do Dynamic Workload Console, consulte:

a seção Guia do Usuário do Console do Dynamic Workload, "Projetando sua Carga de Trabalho".

Usando as Definições de user e streamlogon do Tivoli Workload Scheduler

Nas estações de trabalhoWindows, as definições de usuário são especificadas usando o**composer** no formato **[*workstation*#]*username***. A instância **[*workstation*#]*username*** identifica exclusivamente o usuário Windows no ambiente Tivoli Workload Scheduler. O nome da estação de trabalho é opcional; sua ausência indica que o usuário denominado*username* está definido em todas as estações de trabalho com o Windows na rede do Tivoli Workload Scheduler. Se o usuário denominado*username* estiver definido somente em algumas estações de trabalhoWindows na rede do Tivoli Workload Scheduler, para evitar

inconsistências, você deve criar uma definição de usuário [*workstation*#]*username* para cada estação de trabalho em execução no Windows onde o usuário *username* está definido.

Se você planejar uma tarefa em um agent, em um conjunto ou um conjunto dinâmico, a tarefa é executada com o usuário definido no conjunto ou conjunto dinâmico. Todavia, o usuário deve existir em todas as estações de trabalho no conjunto ou conjunto dinâmico em que planeja executar a tarefa.

Ao definir uma tarefa usando o**composer**, você deve especificar uma estação de trabalho e um logon de usuário válido para a estação de trabalho. O logon é apenas um nome de usuário válido para o Windows, sem o nome da estação de trabalho. Por exemplo, na seguinte definição de tarefa:

```
$JOB
workstation#job01 docommand "dir"
streamlogonusername
```

o valor para `streamlogon` é *username* e não *workstation#username*.

Entretanto, quando você utilizar o comando **altpass**, utilize a definição de usuário no formato

```
workstation#username
```

Para esse comando, é possível omitir o nome da estação de trabalho apenas quando alterar a senha da estação de trabalho a partir da qual o comando está sendo executado.

Usuário de Domínio Confiável

Se o Tivoli Workload Scheduler for ativar tarefas para um usuário de domínio confiável, siga as orientações ao definir as contas de usuário. Pressupondo que o Tivoli Workload Scheduler esteja instalado em `Domain1` para a conta de usuário `maestro` e a conta de usuário `sue` em `Domain2` precise ativar uma tarefa, o seguinte deve ser verdadeiro:

- Deve haver confiança mútua entre `Domain1` e `Domain2`.
- No `Domain1` nos computadores onde os jobs são lançados, `sue` deve ter o direito para **efetuar logon como batch**.
- No `Domain1`, `maestro` deve ser um usuário de domínio.
- Nos controladores de domínio no `Domain2`, `maestro` deve ter o direito para **Acessar este computador a partir da rede**.

Utilizando Definições do Usuário sobre Tipos de Tarefas com Opções Avançadas

No tipos de tarefas com opções avançadas, independente do sistema operacional do agente dinâmico que executará a tarefa, é possível fornecer o nome de usuário de uma definição de usuário na seção de credenciais da tarefa e ter o campo de senha resolvido em tempo de execução com o valor de senha armazenado no banco de dados.

Por exemplo, quando você define a tarefa com o Dynamic Workload Console, você entra com o nome do usuário de uma definição de usuário e clica nas reticências (...) localizadas ao lado do campo de senha para exibir o seguinte widget `Tipo de senha`:

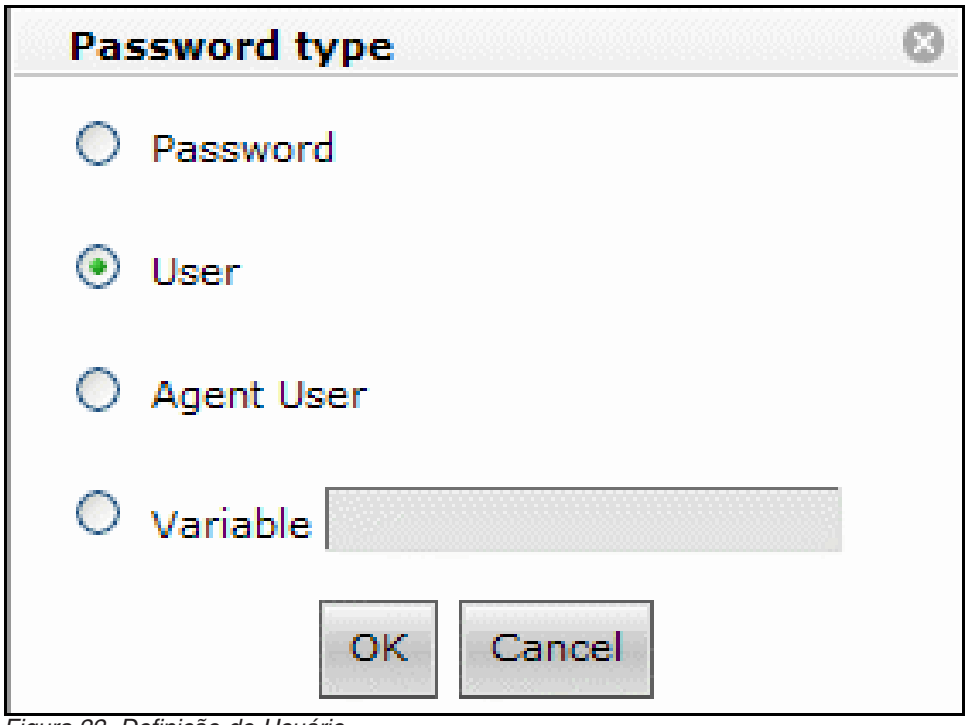


Figura 22. Definição do Usuário

onde você seleciona `Usuário` como mostrado. Também é possível codificar essa opção na seção de tarefa (JSDL) na definição de tarefa no criador. Consulte as seções relacionadas para obter informações adicionais.

Para poder usar essa opção ao definir uma tarefa, você precisa estar autorizado no arquivo de segurança com a palavra-chave de acesso `use` para tipo de objeto`userobj`, que é:

```
userobj access=use
```

O Tivoli Workload Scheduler segue essa sequência quando é chamado para resolver o nome de usuário e a senha no tempo de execução:

- Se a estação de trabalho não for especificada (por exemplo, `${password:myuser}`):
  1. Procura `myuser` na estação de trabalho que executa a tarefa aplicando uma política de distinção entre maiúsculas e minúsculas.
  2. Procura `myuser` na estação de trabalho que executa a tarefa aplicando uma política de distinção entre maiúsculas e minúsculas.
  3. Procura `myuser` sem uma estação de trabalho associada aplicando uma política de distinção entre maiúsculas e minúsculas.
  4. Procura `myuser` sem uma estação de trabalho associada aplicando uma política sem distinção entre maiúsculas e minúsculas.
- Se a estação de trabalho for especificada (por exemplo, `${password:agent#myuser}`):
  1. Procura `myuser` na estação de trabalho`agent` aplicando uma política de distinção entre maiúsculas e minúsculas.
  2. Procura `myuser` na estação de trabalho`agent` aplicando uma política sem distinção entre maiúsculas e minúsculas.
  3. Procura `myuser` sem uma estação de trabalho associada aplicando uma política de distinção entre maiúsculas e minúsculas.

4. Procura `myuser` sem uma estação de trabalho associada aplicando uma política sem distinção entre maiúsculas e minúsculas.

- Se a estação de trabalho for especificada, mas estiver vazia (por exemplo, `${password:#myuser}`):
  1. Procura `myuser` sem uma estação de trabalho associada aplicando uma política de distinção entre maiúsculas e minúsculas.
  2. Procura `myuser` sem uma estação de trabalho associada aplicando uma política sem distinção entre maiúsculas e minúsculas.

Atenção: Definições de usuário com falta de integridade referencial. Isso implica que, se uma definição de usuário referenciada na seção de credenciais de um tipo de tarefa com opções avançadas for alterada ou excluída, nenhum aviso ou mensagem de erro será retornada até que a tarefa seja executada.

Definição do calendário

Um calendário é uma lista de datas que definem se e quando um fluxo de tarefas é executado. Cada definição de calendário tem o seguinte formato e argumentos:

Sintaxe

$calendar
calendarname [“*description*”]
date [...]

[*calendarname* ...]

Argumentos

calendarname
Especifica o nome do calendário. O nome pode conter até oito caracteres alfanuméricos, incluindo traços (-) e sublinhados (_) e deve iniciar com uma letra.

“*description***”**
Fornece uma descrição do calendário. A descrição pode conter até 120 caracteres alfanuméricos. Ela deve ser colocada entre aspas duplas. Pode conter caracteres alfanuméricos, contanto que inicie com uma letra. E pode conter os seguintes caracteres: vírgula (,), ponto (.), traço (-), mais (+), aspas simples (') e igual (=). Ela não pode conter aspas duplas (") que não sejam as de fechamento, dois pontos (:), ponto e vírgula (;)e e comercial (&).

date **[...]**
Especifica uma ou mais datas, separadas por espaços. O formato é *mm/dd/aa*.

Exemplos

O exemplo a seguir define três calendários chamados `monthend`, `paydays` e `holidays`:

```
$calendar
monthend "Month end dates 1st half 2005"
     01/31/2005 02/28/2005 03/31/2005 04/30/2005 05/31/2005 06/30/2005
paydays
     01/15/2005 02/15/2005
     03/15/2005 04/15/2005
     05/14/2005 06/15/2005
feriados
     01/01/2005 02/15/2005 05/31/2005
```

Consulte também

Para obter mais informações sobre como executar a mesma tarefa a partir do Dynamic Workload Console, consulte:

a seção Guia do Usuário do Console do Dynamic Workload, "Projetando sua Carga de Trabalho".

Definição de Variável e Parâmetro

Variáveis e parâmetros são objetos aos quais você designa diferentes valores.

Variáveis e parâmetros são úteis quando você possui valores que mudam dependendo de seus fluxos de tarefas e tarefas. As definições de fluxo de tarefas, de tarefa e de prompt que os usam são atualizadas automaticamente no início do ciclo de produção ou no momento da execução da tarefa, dependendo do formato usado quando a variável é especificada.

Utilize parâmetros como substitutos para valores repetitivos ao definir tarefas e fluxos de tarefas. Por exemplo, o uso de parâmetros para logon de usuário e nomes de arquivo de script em definições de tarefa e para dependências de arquivo e aviso permite o uso de valores que podem ser mantidos centralmente no banco de dados no principal.

Enquanto variáveis estão planejando objetos que são definidos no banco de dados do Tivoli Workload Scheduler e podem ser utilizados por qualquer usuário autorizado no domínio, os parâmetros são definidos e utilizados localmente nos agentes individuais.

As seções a seguir descrevem os procedimentos de migração.

Variáveis

Esses são os parâmetros definidos como objetos de planejamento no banco de dados. Variables utilizando o seguinte comando:

$parm
[*tablename.*]*variablename "variablevalue"*
...

em que:

tablename
: É o nome da tabela de variáveis que deve conter a nova variável. A tabela de variáveis já deve estar definida. Se você não especificar um nome de tabela de variáveis, a variável será incluída na tabela padrão.

variablename
: É o nome da variável. O nome pode conter até 16 caracteres alfanuméricos, incluindo traços (-) e sublinhados (_), e deve iniciar com uma letra.

valor
: É o valor designado à variável. O valor pode conter até 72 caracteres alfanuméricos. Não inclua os nomes de outras variáveis.

Entretanto, a maneira recomendada para definir as variáveis é utilizar um "Definição da Tabela de Variáveis" na página 202. Em qualquer caso, todas as

variáveis são inseridas em uma tabela de variáveis. Se você definir uma variável e não especificar o nome de uma tabela de variáveis, esse nome será incluído na tabela de variáveis padrão.

As variáveis podem ser usadas nas definições de tarefa e de fluxo de tarefas. Elas foram resolvidas, ou seja, elas foram substituídas pelo valor designado quando o plano de produção é gerado ou estendido ou quando você envia uma tarefa ou um fluxo de tarefas para execução. O formato usado para especificar uma variável também determina quando a variável é resolvida com um valor. Os seguintes formatos podem ser usados ao especificar uma variável:

^*variablename***^**
: Especifique a variável neste formato se desejar que ela seja resolvida quando o plano for gerado ou estendido.

${*variablename***}**
: Especifique a variável neste formato se desejar que ela seja resolvida ou sobrescrita quando a tarefa ou fluxo de tarefas for enviado para execução. Uma opção na definição de tarefa que indica para a resolução de variáveis no tempo de execução da tarefa também deve ser especificada. Se esta variável estiver presente somente na tabela de variável padrão, a variável não poderá ser resolvida. Consulte um exemplo de aplicativo deste tipo de variável na seção "Exemplos" na página 201.

Para obter detalhes sobre a resolução de variáveis, consulte "Resolução de Variável" na página 116.

The variable names specified in these definitions are first resolved against variable table definitions and then on local parameters if the variables are not found.

Ao especificar uma variável, coloque a sequência inteira que contém a variável entre aspas.

Se a variável contiver uma parte de um caminho, certifique-se de que os caracteres acento circunflexo não sejam imediatamente precedidos por uma barra invertida (\), pois nesse caso, a seqüência \^ poderia ser interpretada erroneamente como uma seqüência de escape e resolvida pelo analisador como caractere acento circunflexo. Se necessário, mova a barra invertida para a definição da variável entre carets para evitar uma interpretação incorreta da barra invertida. Por exemplo, a tabela a seguir mostra a maneira correta para definir e utilizar uma variável denominada `MYDIR` na tabela de variáveis padrão:

Tabela 45. Como manipular uma barra invertida em substituição de variável.

Errado	Certo

1. Defina a variável `MYDIR` como:
`$PARM`
`MYDIR "scripts"`
2. Utilize-a desta maneira:
`job01 scriptname`
`"c:\operid\^MYDIR^\test.cmd"`
3. Utilize-a desta maneira:
`job01 scriptname`
`"c:\operid\${MYDIR}\test.cmd"`
	1. Defina a variável `MYDIR` como:
	`$PARM`
	`MYDIR "\scripts"`
	2. Utilize-a desta maneira:
	`job01 scriptname`
	`"c:\operid^MYDIR^\test.cmd"`
	3. Utilize-a desta maneira:
	`job01 scriptname`
	`"c:\operid{MYDIR}\test.cmd"`

Isso é verdadeiro para todos os comandos de linha de comandos, interfaces gráficas com o usuário e APIs por meio das quais você utiliza substituição de variável.

Parâmetros

Os parâmetros locais são definidos em um banco de dados local na estação de trabalho em que as tarefas que os utilizam serão executadas. Para defini-los, não utilize este comando **composer** mas o comando do utilitário "parms" na página 520.

Parâmetros locais podem ser utilizados em:
- JCL
- Logon
- Dependências de avisos
- Dependências de arquivos
- Prompts de recuperação

Um parâmetro local é definido nessas palavras-chave ou a partir do script de tarefa chamado utilizando a seguinte sintaxe:

```
'bin\parms PARAMETERNAME'
```

Os parâmetros locais são resolvidos utilizando as definições armazenadas no banco de dados PARMS local da seguinte forma:

- No tempo de execução, na estação de trabalho onde ocorre o processamento da tarefa.
- No tempo de envio, na estação de trabalho na qual a tarefa ou o fluxo de tarefas é enviado a partir da linha de comandos **conman**. A Tabela 46 resume em que palavra-chave do comando **submit** é possível utilizar parâmetros.

Tabela 46. Palavras-chave que podem utilizar parâmetros locais em comandos **submit**

Palavra-chave	submit docommand (comando sbd)	submit file (comando sbf)	submit job (comando sbj)	submit job stream (comando sbs)
abendprompt	✓	✓	✓	
scriptname		✓		
docommand	✓			
logon	✓	✓		
opens	✓	✓	✓	✓
prompt	✓	✓	✓	✓

Para obter informações adicionais sobre como enviar tarefas e fluxos de tarefas na produção a partir da linha de comandos **conman**, consulte o Capítulo 11, "Gerenciando Objetos no Plano - conman", na página 339.

No UNIX, quando você define uma tarefa ou fluxo de tarefa no banco de dados, você deve colocar a cadeia

path/parms *parametername*

entre os caracteres ' ' para assegurar que o parâmetro seja resolvido no tempo de execução na estação de trabalho mesmo se um parâmetro com o mesmo nome estiver definido como um parâmetro global no banco de dados do Tivoli Workload Scheduler. Por exemplo, se você incluir no banco de dados a seguinte definição de tarefa:

```
$jobs
myjob
  docommand "ls ^MYDIR^"
  streamlogon "^MYUSER^"
```

e dois parâmetros denominados `MYDIR` e `MYUSER` são definidos no banco de dados; em seguida, à medida que o plano de produção é criado ou estendido, os dois parâmetros são resolvidos utilizando as definições contidas no banco de dados e seus valores correspondentes são transportados com o arquivo `Symphony`. Se você definir `myjob` no banco de dados da seguinte forma:

```
$jobs
myjob
  docommand "ls 'bin/parms MYDIR'"
  streamlogon "'bin/parms MYUSER'"
```

então, à medida que o plano de produção for criado ou estendido, a única ação que será desempenhada nos dois parâmetros na definição de `myjob` será a remoção dos caracteres ' ', os parâmetros serão transportados no arquivo `Symphony` não resolvido e, então, serão resolvidos no tempo de execução localmente na estação de trabalho de destino utilizando o valor armazenado no banco de dados PARMS.

Exemplos

Dois parâmetros, `glpah` e `gllogon`, são definidos como a seguir:

```
$parm
glpath  "/glfiles/daily"
gllogon  "gluser"
```

Os parâmetros `glpath` e `gllogon` são utilizados no job `gljob2` do fluxo de job `glsched`:

```
schedule glsched on weekdays
:
gljob2
     scriptname "/usr/gl^glpath^"
     streamlogon "^gllogon^"
     opens "^glpath^/datafile"
     prompt ":^glpath^ started by ^gllogon^"
end
```

Um exemplo de uma variável utilizada com a palavra-chave **docommand** é:

```
docommand "ls ^MY_HOME^"
```

O exemplo a seguir demonstra como especificar variáveis em formatos diferentes permite que as variáveis tenham valores diferentes, porque elas são resolvidas em momentos diferentes. Também demonstra como as variáveis podem ser transmitidas de tarefa para tarefa em um fluxo de tarefas. A variável, *SWITCH_VAR* é definida na tabela de variáveis, STATETABLE, com um valor padrão inicial de `on`. A tarefa UPDATE1 é responsável por alterar o valor da variável *SWITCH_VAR* na tabela de variáveis STATETABLE para `off`. O fluxo de tarefas PROCJS contém duas tarefas idênticas, PROC1 e PROC2, nas quais a variável *SWITCH_VAR* foi especificada em dois formatos diferentes. O primeiro configura a variável com o símbolo de acento cincunflexo (^) ^*var_name*^, e o segundo usa o formato ${*var_name*}:

```
<jsdle:script>echo ^SWITCH_VAR^:${SWITCH_VAR}</jsdle:script>
```

A ordem na qual estas tarefas são executadas é a seguinte:

```
SCHEDULE NC117126#PROCJS
VARTABLE STATETABLE
:
NC117126_1# PROC1

NC117126_1# PROC2
 FOLLOWS UPDATE1

NC117126_1# UPDATE1
 FOLLOWS PROC1
END
```

Quando a tarefa for incluída no plano, *SWITCH_VAR*, definido em PROC1 e PROC2, considerará imediatamente o valor padrão designado na tabela de variáveis, `on`. Quando o fluxo de tarefas for enviado para execução, a primeira tarefa a ser enviada será PROC1 e a variável definida como *SWITCH_VAR* será resolvida como `on` para que as variáveis na tarefa PROC1 sejam resolvidas como:

```
<jsdle:script>echo on:onjsdle:script>echo on:on</jsdle:script>
```

UPDATE1 executa então a configuração do valor de *SWITCH_VAR* na tabela de variáveis como `off` para que, quando PROC2 for executado, as variáveis sejam resolvidas como:

```
<jsdle:script>echo on:onjsdle:script>echo on:off</jsdle:script>
```

A variável especificada como ^*SWITCH_VAR*^ na tarefa mantém o valor de `on`, porque as variáveis neste formato são resolvidas quando o fluxo de tarefas é incluído no plano e não são atualizadas quando a tarefa é enviada para execução. Em vez disso, a variável especificada no formato *${SWITCH_VAR}*, que foi configurado anteriormente como `on`, agora é atualizada com o novo valor na tabela de variáveis `off`.

Criando uma Definição de Variável Usando oDynamic Workload Console

Para criar uma definição de variável no Dynamic Workload Console, você deve incluí-la em uma definição de tabela de variáveis:

1. Clique em **Tivoli Workload Scheduler→Carga de Trabalho→Design→Criar Definições de Carga de Trabalho**
2. Selecione um nome de mecanismo e clique em **Ir**
3. Abra no modo de edição uma tabela de variáveis existente a partir da área de janela Abertura Rápida ou crie uma nova tabela de variáveis, conforme descrito em "Definição da Tabela de Variáveis"
4. No painel Propriedades - Tabela de Variável, clique na guia Variáveis e inclua novas definições de variáveis clicando no ícone "+" (Incluir) e especificando nomes e valores de variáveis

Para obter informações adicionais, consulte a Capítulo 6, "Customizando a Carga de Trabalho Utilizando Tabelas de Variáveis", na página 113.

Definição da Tabela de Variáveis

Uma tabela de variáveis é um objeto que agrupa várias variáveis. Todos os parâmetros globais (agora denominados *variáveis*) que você utiliza no planejamento da carga de trabalho estão contidos em pelo menos uma tabela de variáveis. As seguintes opções estão disponíveis:

- Defini-las quando definir uma tabela de variáveis da maneira descrita aqui. Esta é a maneira recomendada.
- Defini-las individualmente com o comando **composer $parm** no formato [*tablename.*]*variablename* "*variablevalue*" . Se você não especificar um nome de tabela, a nova variável será colocada na tabela de variáveis padrão.

Você não é forçado a criar tabelas de variáveis para poder criar e utilizar variáveis. É possível nunca criar uma tabela e nunca utilizar uma explicitamente. Mas o planejador vem com uma padrão e sempre que você cria ou gerencia uma variável sem denominar a tabela, ele a armazena ou procura por ela lá.

Porém, é possível definir mais de uma variável com o mesmo nome, mas valor diferente, e colocá-las em tabelas diferentes. Utilizando as tabelas de variáveis, você designa valores diferentes para a mesma variável e, dessa forma, reutiliza a mesma variável em definições de tarefas e ao definir dependências de aviso e de arquivo. Tabelas de variáveis podem ser designadas no ciclo de execução, fluxo de tarefas e nível da estação de trabalho.

As tabelas de variáveis podem ser particularmente úteis em definições de tarefas quando uma definição de tarefa é utilizada como um modelo para uma tarefa que pertence a mais de um fluxo de tarefas. Por exemplo, é possível designar valores diferentes para a mesma variável e reutilizar a mesma definição de tarefa em fluxos de tarefas diferentes.

Para obter informações adicionais, consulte a Capítulo 6, “Customizando a Carga de Trabalho Utilizando Tabelas de Variáveis”, na página 113.

Sintaxe

vartable *tablename*
[**description** *“description”*]
[**isdefault**]
members
[*variablename “variablevalue”*]
...
[*variablename “variablevalue”*]
end

Argumentos

vartable *tablename*
O nome da tabela de variáveis. O nome deve iniciar com uma letra e pode conter caracteres alfanuméricos, traços e sublinhados. Ela pode conter até 80 caracteres.

description "*tabledescription*"
A descrição da tabela de variáveis. O texto deve ser colocado entre aspas duplas. A descrição pode conter até 120 caracteres alfanuméricos. Ela não pode conter aspas duplas (") que não sejam as de fechamento, dois pontos (:), ponto e vírgula (;) e e comercial (&).

isdefault
Quando especificada, a tabela é a tabela padrão. Você não pode marcar mais de uma tabela como tabela padrão. Quando você marca uma tabela de variáveis como a tabela padrão, a tabela de variáveis atual não é mais a tabela padrão. Ao migrar o banco de dados de uma versão anterior, o produto cria a tabela de variáveis padrão com todas as variáveis já definidas.

members *variablename* "*variablevalue*"
A lista de variáveis e seus valores separados por espaços. O nome pode conter até 16 caracteres alfanuméricos, incluindo traços (-) e sublinhados (_), e deve iniciar com uma letra. O valor pode conter até 72 caracteres alfanuméricos. Os valores devem ser colocados entre aspas duplas.

Exemplo

O exemplo a seguir mostra uma tabela de variáveis e seu conteúdo.

```
VARTABLE TEST1
  MEMBERS
   DEVBATCH "DOMD\IMSBATCH\SAME"
   PARAM_01 "date"
   PARAM_02 "root"
   PARM_01 "PARM_001"
   PRPT_02 "PARM_002"
   PRPT_03 "PARM_003"
   PRPT_04 "PARM_004"
   PRPT_05 "PARM_005"
```

```
   SAME17 "test/for/variable âwith samename > variable/table"
   SLAV10 "/nfsdir/billingprod/crmb/MAESTRO_JOB/AG82STGGDWHSCART"
   SLAV11 "/nfsdir/billingprod/crmb/MAESTRO_JOB/AG82CDMGALLBCV"
   SLAV12 "/nfsdir/billingprod/crmb/MAESTRO_JOB/AG82CDMGRISCTRAF"
   SLAV13 "/opt/crm/DWH_OK/Businness_Copy_ok"
   SLAV14 "/opt/crm/DWH_OK/DW_Canc_Cust_Gior_ok_"
   TRIGGER "/usr/local/samejobtriggers"
   VFILE2 "testforvarwithsamename2.sh"
   VUSER2 "same_user2"
   WRAPPER "/usr/local/sbin/same/phi_job.ksh"
END
```

Considerações do Arquivo de Segurança

Do ponto de vista das autorizações do arquivo de segurança, a permissão para agir nas entradas de variável contidas em uma tabela depende da permissão geral concedida na tabela, como mostrado a seguir.

Tabela 47. Palavra-chave de Acesso Necessárias em Tabela de Variável o Arquivo de Segurança (Objeto vartable) e Ações Permitidas.

Palavra-chave obrigatória de acesso ao arquivo de segurança na tabela de variáveis incluída	Ação permitida nas entradas de variável listadas
Modificar	Incluir
	Excluir
	Modificar
	Renomear
Exibir	Exibir
Desbloquear	Desbloquear

Consulte também

Para obter mais informações sobre como executar a mesma tarefa a partir do Dynamic Workload Console, consulte:

a seção Guia do Usuário do Console do Dynamic Workload, "Projetando sua Carga de Trabalho".

Definição do Prompt

Um aviso identifica uma mensagem textual que é exibida para o operador e pára o processamento da tarefa ou do fluxo de tarefas até que uma resposta afirmativa seja recebida (manualmente do operador ou automaticamente por uma ação da regra de evento). Após o aviso ser respondido, o processamento continua. Os prompts podem ser utilizados como dependências em jobs e fluxos de job. É possível utilizar variáveis em prompts.

Existem dois tipos de prompts:

prompts locais ou não denominados
: Um prompt não denominado é um prompt definido em uma definição de tarefa ou de fluxo de tarefas utilizando a apalavra-chave **prompt**, não tem nenhum nome designado e não está definido como um objeto de planejamento no banco de dados, portanto, não pode ser utilizado por outras tarefas ou fluxos de tarefas.

prompts globais ou denominados
: Um prompt global é definido no banco de dados como um objeto de planejamento, ele é identificado por um nome exclusivo e pode ser utilizado por qualquer tarefa ou fluxo de tarefa. As variáveis em prompts

globais são determinadas sempre utilizando a tabela de variáveis padrão. Isso ocorre porque o prompt global é utilizado por todas as tarefas e fluxos de tarefas, portanto, apenas um valor deve ser utilizado para a resolução da variável.

Esta seção descreve prompts globais. Para obter informações adicionais sobre prompts locais, consulte "Tarefa" na página 711 e "Definição de fluxo de tarefas" na página 216.

Nota: As definições de prompt predefinido ou global são reconfiguradas toda vez que a tarefa **JnextPlan** é executada.

Sintaxe

$prompt
promptname "[**:** | **!**]*text*"

[*promptname* ...]

Argumentos

promptname

Especifica o nome do prompt. O nome pode conter até 8 caracteres alfanuméricos, incluindo traços (-) e sublinhados (_), e deve iniciar com uma letra.

text

Fornece o texto do prompt. O texto do prompt pode conter até duzentos caracteres alfanuméricos. Baseado no caractere que antecede o texto, o prompt pode se comportar de maneira diferente.

- Se o texto começar com dois pontos (:), o prompt será exibido, mas não será necessária nenhuma resposta para continuar o processamento.
- Se o texto começar com um ponto de exclamação (!), o prompt será exibido, mas não será gravado no arquivo de log.

É possível utilizar um ou mais parâmetros como toda a cadeia de texto ou parte dela para um prompt. No caso de utilizar um parâmetro, a cadeia de parâmetros deverá ser colocada entre acentos circunflexos (^). Consulte "Definição de Variável e Parâmetro" na página 198 para obter um exemplo.

Nota: Em prompts locais, os circunflexos (^) que não identificam um parâmetro devem ser precedidos por uma barra invertida (\) para evitar que causem erros no prompt. Em avisos globais, os acentos circunflexos não precisam ser precedidos por uma barra invertida.
Você pode incluir uma barra invertida n (**n**) no texto para criar uma nova linha.

Exemplos

O exemplo a seguir define três prompts:

```
$prompt
     prmt1 "ready for job4? (y/n)"
     prmt2 ":job4 launched"
     prmt3 "!continue?"
```

Consulte também

Para obter mais informações sobre como executar a mesma tarefa a partir do Dynamic Workload Console, consulte:

a seção Guia do Usuário do Console do Dynamic Workload, "Projetando sua Carga de Trabalho".

Definição do recurso

Representam recursos de planejamento físicos ou lógicos que podem ser utilizados como dependências para jobs e fluxos de job. Os recursos podem ser utilizados como dependências somente por tarefas e fluxos de tarefas que são executados na estação de trabalho onde o recurso está definido.

Devido ao mecanismo de resolução de dependências de recursos, uma dependência de recurso no nível do fluxo de tarefas pode ser considerado 'local' (e, então, seu uso suportado) em vez de 'global', quando o fluxo de tarefas e todas as suas tarefas forem definidos na mesma estação de trabalho que o recurso. No entanto, um agente padrão e seu host podem fazer referência aos mesmos recursos. Para obter informações adicionais, consulte a palavra-chave “needs” na página 243.

Sintaxe

$resource
workstation#*resourcename units* [“*description*”]

[*workstation*#*resourcename* ...]

Argumentos

estação de trabalho
Especifica o nome da estação de trabalho ou classe de estação de trabalho na qual o recurso é utilizado.

resourcename
Especifica o nome do recurso. O nome pode conter até oito caracteres alfanuméricos, incluindo traços (-) e sublinhados (_) e deve iniciar com uma letra.

units Especifica o número de unidades de recurso disponíveis. Os valores podem ser de **0** a **1024**.

“*description***”**

Fornece uma descrição do recurso. A descrição pode conter até 120 caracteres alfanuméricos. Ele deve ser colocado entre aspas duplas.

As unidades de recursos envolvidas nas dependências de necessidades para uma tarefa ou para um fluxo de tarefa permanecem ocupadas até a tarefa ou o fluxo de tarefa ser concluído (com êxito ou não). As unidades de recurso são liberadas assim que a tarefa ou o fluxo de tarefa for concluído.

Quando várias tarefas e fluxos de tarefas dependem do mesmo recurso, se não houver unidades de recursos suficientes disponíveis para todos eles, elas serão designadas de acordo com a prioridade da tarefa ou do fluxo de tarefas. O status de uma tarefa ou fluxo de tarefas torna-se READY assim que todas as suas dependências são resolvidas. Se a CPU limite configurada na estação de trabalho

não permitir a execução no momento, aguarda no estado READY. A única exceção para este comportamento é quando a tarefa ou fluxo de tarefa é GO ou HI, nesse caso, ela inicia independentemente do valor configurado para CPU limite.

Exemplos

O exemplo a seguir define quatro recursos:

```
$resource
  ux1#tapes 3 "tape units"
  ux1#jobslots 24 "job slots"
  ux2#tapes 2 "tape units"
  ux2#jobslots 16 "job slots"
```

Consulte também

Para obter mais informações sobre como executar a mesma tarefa a partir do Dynamic Workload Console, consulte:

a seção Guia do Usuário do Console do Dynamic Workload, "Projetando sua Carga de Trabalho".

Definição do Grupo de Ciclos de Execução

Um grupo do ciclo de execução é um objeto de banco de dados no qual um ou mais ciclos de execução são definidos. Os ciclos de execução combinados juntos produzem um conjunto de datas de execução para um fluxo de tarefas. Cada definição do grupo do ciclo de execução possui o seguinte formato e argumentos:

Sintaxe

$runcyclegroup
runcyclegroupname [*"description"*]
vartable*tablename*
[freedays *calendarname* **[-sa]** **[-su]]**
[on **[runcycle** *name*]
[validfrom *date*] **[validto** *date*]
[description
"text"]
[vartable *table_name*]
{*date* | *day* | *calendar* | *request* | *"icalendar"*} **[,...]**
[fdignore | **fdnext** | **fdprev**][**subset** *subsetname* **AND** | **OR**]
[({**at** *time* [+*n* **day**[**s**]] |
schedtime *time* [+*n* **day**[**s**]]}
[until *time* [+*n* **day**[**s**]] **[onuntil** *action*]]
[deadline *time* [+*n* **day**[**s**]]])]
[,...]
[except [**runcycle** *name*]
[validfrom *date*] **[validto** *date*]
[description
"text"]
{*date* | *day* | *calendar* | *request* | *"icalendar"*} **[,...]**
[fdignore | **fdnext** | **fdprev**][**subset** *subsetname* **AND** | **OR**]
[{(**at** *time* [+*n* **day**[**s**]])] |
(**schedtime** *time* [+*n* **day**[**s**]])}]
[,...]
[{**at** *time* [***timezone*** | ***tz*** *tzname*] [+*n* **day**[**s**]] |
schedtime *time* [***timezone*** | ***tz*** *tzname*] [+*n* **day**[**s**]]}]

[**until** *time* [**timezone**|**tz** *tzname*] [+*n* **day**[**s**]] [**onuntil** *action*]]
[**deadline** *time* [**timezone**|**tz** *tzname*] [+*n* **day**[**s**]]]
end

Argumentos

runcyclegroupname
Especifica o nome do grupo do ciclo de execução. O nome pode conter até oito caracteres alfanuméricos, incluindo traços (-) e sublinhados (_) e deve iniciar com uma letra.

"*description***"**
Fornece uma descrição do grupo do ciclo de execução. A descrição pode conter até 120 caracteres alfanuméricos. Ela deve ser colocada entre aspas duplas. Pode conter caracteres alfanuméricos, contanto que inicie com uma letra. E pode conter os seguintes caracteres: vírgula (,), ponto (.), traço (-), mais (+), aspas simples (') e igual (=). Ela não pode conter aspas duplas (") que não sejam as de fechamento, dois pontos (:), ponto e vírgula (;) e e comercial (&).

vartable *tablename*
O nome da tabela de variáveis. O nome deve iniciar com uma letra e pode conter caracteres alfanuméricos, traços e sublinhados. Ela pode conter até 80 caracteres.

[freedays *Calendar_Name* **[-sa] [-su]]**
Especifica o calendário de tempo livre para calcular dias úteis para o fluxo de tarefas. Ele também pode configurar sábados e domingos como dias úteis.

Calendar_Name
O nome do calendário que deve ser utilizado como o calendário de dias livres para o fluxo de tarefa. Se *Calendar_Name* não estiver no banco de dados, o Tivoli Workload Scheduler emitirá uma mensagem de aviso quando você salvar o fluxo de tarefas. Se *Calendar_Name* não estiver no banco de dados quando o comando **schedulr** for executado, o Tivoli Workload Scheduler emitirá uma mensagem de erro e utilizará o calendário padrão **holidays** no lugar. Não utilize os nomes de dias da semana para nomes de calendário.

-sa Sábados são *dias úteis*.

-su Domingos são *dias úteis*.

Consulte “dias livres” na página 235 para obter detalhes e exemplos.

runcycle *name*
Especifica uma etiqueta com um nome amigável para o ciclo de execução especificado nas linhas a seguir.

valid from *date* **... valid to***date*
Delimita o espaço de tempo durante o qual o fluxo de tarefas está ativo, ou seja, o fluxo de tarefas é incluído no plano de produção. Observe que a data especificada como o valor **valid to** não está incluída no ciclo de execução; portanto, nessa data o fluxo de tarefas não está ativo.

description "*text***"**
Contém uma descrição do ciclo de execução.

vartable
Especifica o nome da tabela de variáveis a ser utilizada pelo ciclo de execução.

date Especifica um ciclo de execução que é executado em datas específicas. A sintaxe utilizada para esse tipo é:

yyyymmdd [,yyyymmdd][,...]Por exemplo, para um fluxo de tarefas que é planejado para execução em 25 de maio de 2009 e em 12 de junho de 2009, o valor é:

```
on
20090525,20090612
```

day Especifica um ciclo de execução que é executado em dias específicos. A sintaxe utilizada para esse tipo é:

{mo | tu | we | th | fr | sa | su}Por exemplo, para um fluxo de tarefas que está planejado para execução toda segunda-feira, o valor é:

```
on
mo
```

calendário
As datas especificadas com este nome em um calendário. O nome do calendário pode vir seguido de um deslocamento no seguinte formato:

{**+** | **-**}*n* {**day[s]** | **weekday[s]** | **workday[s]**}

Em que:

n O número de dias, dias da semana (menos sábado e domingo) ou dias úteis.

days Todos os dias da semana.

weekdays
Todos os dias da semana, exceto sábado e domingo.

dias úteis
Todos os dias da semana, exceto para Sábados e Domingos (a menos que seja especificado de outra forma com a palavra-chave **freedays**) e para as datas marcadas em um calendário de dias livres designado ou no calendário **holidays**.

request Seleciona o fluxo de tarefas somente quando solicitado. Isto é utilizado para fluxos de job que são selecionados por nome em vez de data. Para evitar que um fluxo de tarefa planejado seja selecionado para **JnextPlan**, altere sua definição para ON REQUEST.

Nota: Ao tentar executar um fluxo de tarefas que contenha horários "on request", considere que:

- "On request" sempre antecede "at".
- "On request" nunca antecede "on".

icalendar
Representa um padrão utilizado para especificar uma regra recorrente que descreve quando um fluxo de tarefa é executado.

A sintaxe utilizada para o ciclo de execução com tipo *icalendar* é o seguinte:

FREQ={DAYLY | WEEKLY | MONTHLY | YEARLY}

[;INTERVAL=[-]*n*]

[;{BYFREEDAY | BYWORKDAY | BYDAY=*weekday_list* |

BYMONTHDAY=*monthday_list***}]**

em que o valor padrão para a palavra-chave **INTERVAL** é 1.

Utilizando *icalendar* é possível especificar que um fluxo de tarefa seja executado:

a cada *n* dias
utilizando o seguinte formato:

FREQ=DAILY[;INTERVAL=*n*]

em que o valor configurado para **valid from** é o primeiro dia das datas resultantes.

Por exemplo, para um fluxo de tarefas que está planejado para execução diariamente, o valor é:

```
FREQ=DAILY
```

Para um fluxo de tarefa que está planejado para execução a cada dois dias, o valor é:

```
FREQ=DAILY;INTERVAL=2
```

todo dia livre ou útil
utilizando o seguinte formato:

FREQ=DAILY[;INTERVAL=*n*]

;BYFREEDAY | BYWORKDAY

Por exemplo, para um fluxo de tarefa que está planejado para executar todo dia livre o valor é:

```
FREQ=DAILY;BYFREEDAY
```

Para um fluxo de tarefa que está planejado para execução a cada dois dias úteis, o valor é:

```
FREQ=DAILY;INTERVAL=2;BYWORKDAY
```

a cada *n* semanas em *weekdays* específicos
utilizando o seguinte formato:

FREQ=WEEKLY[;INTERVAL=*n*]

;BYDAY=*weekday_list*

em que o valor configurado para *weekday_list* é

```
[SU][,MO][,TU][,WE][,TH][,FR][,SA]
```

Por exemplo, para um fluxo de tarefas que está planejado para execução toda semana na sexta-feira e no sábado, o valor é:

```
FREQ=WEEKLY;BYDAY=FR,SA
```

Para um fluxo de tarefa que está planejado para execução a cada três semanas na sexta-feira, o valor é:

```
FREQ=WEEKLY;INTERVAL=3;BYDAY=FR
```

a cada *n* meses em datas específicas do mês
utilizando o seguinte formato:

FREQ=MONTHLY[;INTERVAL=*n*]

;BYMONTHDAY=*monthday_list*

em que o valor configurado para *monthday_list* é representado por uma lista de

```
[+number_of_day_from_beginning_of_month]
[-number_of_day_from_end_of_month]
[number_of_day_of_the_month]
```

Por exemplo, para um fluxo de tarefas que está planejado para execução mensal no 27º dia, o valor é:

```
FREQ=MONTHLY;BYMONTHDAY=27
```

Para um fluxo de tarefa que está planejado para execução a cada seis meses no 15º e no último dia do mês, o valor é:

```
FREQ=MONTHLY;INTERVAL=6;BYMONTHDAY=15,-1
```

a cada *n* **meses em dias específicos de semanas específicas**
utilizando o seguinte formato:

FREQ=MONTHLY[;INTERVAL=*n*]

;BYDAY=day_of_m_week_list

em que o valor configurado para *day_of_m_week_list* é representado por uma lista de

```
[+number_of_week_from_beginning_of_month]
[-number_of_week_from_end_of_month]
[weekday]
```

Por exemplo, para um fluxo de tarefas que está planejado para execução mensal na primeira segunda-feira e na última sexta-feira, o valor é:

```
FREQ=MONTHLY;BYDAY=1MO,-1FR
```

Para um fluxo de tarefa que está planejado para execução a cada seis meses na 2ª sexta-feira, o valor é:

```
FREQ=MONTHLY;INTERVAL=6;BYDAY=2TU
```

a cada *n* **anos**
utilizando o seguinte formato:

FREQ=YEARLY[;INTERVAL=*n*]

em que o valor configurado para **valid from** é o primeiro dia das datas resultantes.

Por exemplo, para um fluxo de tarefas que está planejado para execução anualmente, o valor é:

```
FREQ=YEARLY
```

Para um fluxo de tarefas planejado para execução a cada dois anos, o valor é:

```
FREQ=YEARLY;INTERVAL=2
```

fdignore | fdnext | fdprev
Indica a regra a ser aplicada se a data selecionada para executar a tarefa ou fluxo de tarefa cair em um dia livre. As configurações disponíveis são:

fdignore
Não incluir a data.

fdnext Inclua o dia útil mais próximo após o dia livre.

fdprev
Inclua o dia útil mais próximo antes do dia livre.

[subset *subsetname* **AND | OR]**

subset *subsetname*
Especifica o nome do subconjunto. Se não for especificado um nome, SUBSET_1 será usado por padrão.

AND | OR
Por padrão, os ciclos de execução em um subconjunto estão em um

relacionamento **OR** lógico, mas é possível alterá-lo para um **AND** lógico, desde que o resultado do grupo do ciclo de execução seja uma data ou conjunto de dados positivo (Inclusive).

runcycle *name*
Especifica uma etiqueta com um nome amigável para o ciclo de execução especificado nas linhas a seguir.

valid from *date* **... valid to***date*
Delimita o espaço de tempo durante o qual o fluxo de tarefas está ativo, ou seja, o fluxo de tarefas é incluído no plano de produção. Observe que a data especificada como o valor **valid to** não está incluída no ciclo de execução; portanto, nessa data o fluxo de tarefas não está ativo.

description ***"****text****"***
Contém uma descrição do ciclo de execução.

date Especifica um ciclo de execução que é executado em datas específicas. A sintaxe utilizada para esse tipo é:

yyyymmdd [,yyyymmdd][,...]Por exemplo, para um fluxo de tarefas que é planejado para execução em 25 de maio de 2009 e em 12 de junho de 2009, o valor é:

```
on
20090525,20090612
```

day Especifica um ciclo de execução que é executado em dias específicos. A sintaxe utilizada para esse tipo é:

{mo | tu | we | th | fr | sa | su}Por exemplo, para um fluxo de tarefas que está planejado para execução toda segunda-feira, o valor é:

```
on
mo
```

calendário
As datas especificadas com este nome em um calendário. O nome do calendário pode vir seguido de um deslocamento no seguinte formato:

{**+** | **-**}*n* {**day[s]** | **weekday[s]** | **workday[s]**}

Em que:

n O número de dias, dias da semana (menos sábado e domingo) ou dias úteis.

days Todos os dias da semana.

weekdays
Todos os dias da semana, exceto sábado e domingo.

dias úteis
Todos os dias da semana, exceto para Sábados e Domingos (a menos que seja especificado de outra forma com a palavra-chave **freedays**) e para as datas marcadas em um calendário de dias livres designado ou no calendário **holidays**.

request Seleciona o fluxo de tarefas somente quando solicitado. Isto é utilizado para fluxos de job que são selecionados por nome em vez de data. Para evitar que um fluxo de tarefa planejado seja selecionado para **JnextPlan**, altere sua definição para ON REQUEST.

Nota: Ao tentar executar um fluxo de tarefas que contenha horários "on request", considere que:

- "On request" sempre antecede "at".

- "On request" nunca antecede "on".

icalendar

Representa um padrão utilizado para especificar uma regra recorrente que descreve quando um fluxo de tarefa é executado.

A sintaxe utilizada para o ciclo de execução com tipo *icalendar* é o seguinte:

FREQ={DAYLY | WEEKLY | MONTHLY | YEARLY}

[;INTERVAL=[-]*n*]

[;{BYFREEDAY | BYWORKDAY | BYDAY=*weekday_list* |

BYMONTHDAY=*monthday_list*}**]**

em que o valor padrão para a palavra-chave **INTERVAL** é 1.

Utilizando *icalendar* é possível especificar que um fluxo de tarefa seja executado:

a cada *n* **dias**
utilizando o seguinte formato:

FREQ=DAILY[;INTERVAL=*n*]

em que o valor configurado para **valid from** é o primeiro dia das datas resultantes.

Por exemplo, para um fluxo de tarefas que está planejado para execução diariamente, o valor é:

```
FREQ=DAILY
```

Para um fluxo de tarefa que está planejado para execução a cada dois dias, o valor é:

```
FREQ=DAILY;INTERVAL=2
```

todo dia livre ou útil
utilizando o seguinte formato:

FREQ=DAILY[;INTERVAL=*n*]

;BYFREEDAY | BYWORKDAY

Por exemplo, para um fluxo de tarefa que está planejado para executar todo dia livre o valor é:

```
FREQ=DAILY;BYFREEDAY
```

Para um fluxo de tarefa que está planejado para execução a cada dois dias úteis, o valor é:

```
FREQ=DAILY;INTERVAL=2;BYWORKDAY
```

a cada *n* **semanas em** *weekdays* **específicos**
utilizando o seguinte formato:

FREQ=WEEKLY[;INTERVAL=*n*]

;BYDAY=*weekday_list*

em que o valor configurado para *weekday_list* é

```
[SU][,MO][,TU][,WE][,TH][,FR][,SA]
```

Por exemplo, para um fluxo de tarefas que está planejado para execução toda semana na sexta-feira e no sábado, o valor é:

```
FREQ=WEEKLY;BYDAY=FR,SA
```

Para um fluxo de tarefa que está planejado para execução a cada três semanas na sexta-feira, o valor é:

```
FREQ=WEEKLY;INTERVAL=3;BYDAY=FR
```

a cada *n* **meses em datas específicas do mês**
utilizando o seguinte formato:

FREQ=MONTHLY[;INTERVAL=*n*]

;BYMONTHDAY=*monthday_list*

em que o valor configurado para *monthday_list* é representado por uma lista de

```
[+number_of_day_from_beginning_of_month]
[-number_of_day_from_end_of_month]
[number_of_day_of_the_month]
```

Por exemplo, para um fluxo de tarefas que está planejado para execução mensal no 27º dia, o valor é:

```
FREQ=MONTHLY;BYMONTHDAY=27
```

Para um fluxo de tarefa que está planejado para execução a cada seis meses no 15º e no último dia do mês, o valor é:

```
FREQ=MONTHLY;INTERVAL=6;BYMONTHDAY=15,-1
```

a cada *n* **meses em dias específicos de semanas específicas**
utilizando o seguinte formato:

FREQ=MONTHLY[;INTERVAL=*n*]

;BYDAY=*day_of_m_week_list*

em que o valor configurado para *day_of_m_week_list* é representado por uma lista de

```
[+number_of_week_from_beginning_of_month]
[-number_of_week_from_end_of_month]
[weekday]
```

Por exemplo, para um fluxo de tarefas que está planejado para execução mensal na primeira segunda-feira e na última sexta-feira, o valor é:

```
FREQ=MONTHLY;BYDAY=1MO,-1FR
```

Para um fluxo de tarefa que está planejado para execução a cada seis meses na 2ª sexta-feira, o valor é:

```
FREQ=MONTHLY;INTERVAL=6;BYDAY=2TU
```

a cada *n* **anos**
utilizando o seguinte formato:

FREQ=YEARLY[;INTERVAL=*n*]

em que o valor configurado para **valid from** é o primeiro dia das datas resultantes.

Por exemplo, para um fluxo de tarefas que está planejado para execução anualmente, o valor é:

```
FREQ=YEARLY
```

Para um fluxo de tarefas planejado para execução a cada dois anos, o valor é:

```
FREQ=YEARLY;INTERVAL=2
```

fdignore | fdnext | fdprev
Especifica uma regra que deve ser aplicada quando a data selecionada para exclusão cai em um dia livre. Pode ser um dos seguintes:

fdignore
Não excluir a data.

fdnext Exclua o dia útil mais próximo após o dia livre.

fdprev
Exclua o dia útil mais próximo antes do dia livre.

subset *subsetname*
Especifica o nome do subconjunto. Se não for especificado um nome, SUBSET_1 será usado por padrão.

AND | OR
Por padrão, os ciclos de execução em um subconjunto estão em um relacionamento **OR** lógico, mas é possível alterá-lo para um **AND** lógico, desde que o resultado do grupo do ciclo de execução seja uma data ou conjunto de dados positivo (Inclusive).

Exemplo

O exemplo a seguir define um grupo de ciclos de execução denominado RCG2, que contém um ciclo de execução inclusivo, RUN_CYCLE1, e dois ciclos de execução exclusivos, RUN_CYCLE2 e RUN_CYCLE3. Para determinar o planejamento de execução do fluxo de tarefas associado a esse grupo de ciclos de execução, a intersecção dos dois ciclos de execução exclusivos (os dois ciclos de execução exclusivos possuem um relacionamento AND lógico entre eles) é subtraída do ciclo de execução inclusivo. A seguir há as características do grupo de ciclos de execução:

Um ciclo de execução inclusivo RUN_CYCLE1
em que,

- O calendário, CAL1, define os dias que devem ser considerados como dias de folga para o fluxo de tarefas. Sábado e Domingo são declarados como dias úteis.
- O fluxo de tarefas é executado não antes de dois dias após 31 de março de 2008 (2 de abril), e não depois de dois dias após 12 de abril de 2008 (14 de abril). A cada dia, o fluxo de tarefas é atrasado em dois dias.
- Os fluxos de tarefas são executados todos os dias (após o atraso de dois dias) iniciando às 7h e não pode iniciar após às 9h, caso contrário, ele será suprimido e não será executado de forma alguma. O fluxo de tarefas deverá ser concluído até às 10h.

Um ciclo de execução exclusivo, RUN_CYCLE2
Se o fluxo de tarefas cair em um dia de folga, então o dia útil mais próximo antes do dia de folga será excluído.

Se o fluxo de tarefas cair em 1º de abril de 2008, e se esse dia for um dia de folga, então o dia útil mais próximo após o dia de folga será excluído.

Um ciclo de execução exclusivo, RUN_CYCLE3
Se o fluxo de tarefas cair em 1º de abril de 2008, e se esse dia for um dia de folga, então o dia útil mais próximo após o dia de folga será excluído.

```
RUNCYCLEGROUP RCG2
DESCRIPTION "Sample RunCycle Group"
VARTABLE TABLE1
FREEDAYS CAL1 -SA -SU
  ON  RUNCYCLE RUN_CYCLE1 VALIDFROM 03/31/2008 VALIDTO 04/12/2008 DESCRIPTION
        "Inclusive Run Cycle" VARTABLE TABLE1 "FREQ=DAILY;" FDIGNORE
        (AT 0700 +2 DAYS UNTIL 0900 +2 DAYS ONUNTIL SUPPR DEADLINE 1000 +2 DAYS)

  EXCEPT RUNCYCLE RUN_CYCLE2 VALIDFROM 03/31/2008 VALIDTO 04/12/2008 DESCRIPTION
        "Exclusive Run Cycle" CAL1 FDPREV SUBSET SUBSET_A AND
        (AT 0700 +2 DAYS)

  EXCEPT RUNCYCLE RUN_CYCLE3 VALIDFROM 03/31/2008 VALIDTO 04/12/2008 DESCRIPTION
        "Exclusive Run Cycle" 04/01/2008 FDNEXT SUBSET SUBSET_A AND
        (SCHEDTIME 0700 +2 DAYS)
SCHEDTIME 0700 TZ Europe/Berlin +2 DAYS UNTIL 0900 TZ Europe/Berlin +2 DAYS ONUNTIL
         CONT DEADLINE 1000 TZ Europe/Berlin +2 DAYS
END
```

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

o Guia do Usuário do Console do Dynamic Workload, seção Criando Definições de Fluxo de Tarefas.

Definição de fluxo de tarefas

Um fluxo de tarefa consiste em uma seqüência de tarefas a serem executadas, juntamente com as dependências de horas, prioridades e outras dependências que determinam a ordem do processamento.

Um fluxo de tarefas começa com uma palavra-chave **schedule** seguida de atributos e dependências. O delimitador de dois pontos introduz as tarefas chamadas pelo fluxo de tarefas. Cada job possui seus próprios atributos e dependências.

Sintaxe

schedule [*workstation*#]*jobstreamname*
  # *comment*
  [**validfrom** *date*]
  [**timezone** | **tz** *tzname*]
  [**description** *"text"*]
  [**draft**]
  [**vartable** *table_name*]
  [**freedays** *calendarname* [**-sa**] [**-su**]]
  [**on** [**runcycle** *name*]
    [**validfrom** *date*] [**validto** *date*]
    [**description** *"text"*]
    [**vartable** *table_name*]
   {*date* | *day* | *calendar* | *request* | *"icalendar"* | *runcyclegroup*} [**,...**]
    [**fdignore** | **fdnext** | **fdprev**]
    [({**at** *time* [+*n* **day**[**s**]] |
    schedtime *time* [+*n* **day**[**s**]]}
    [**until** *time* [+*n* **day**[**s**]] [**onuntil** *action*]]
    [**deadline** *time* [+*n* **day**[**s**]]])]]
  [**,...**]
  [**except** [**runcycle** *name*]
      [**validfrom** *date*] [**validto** *date*]
      [**description** *"text"*]
      {*date* | *day* | *calendar* | *request* | *"icalendar"* | *runcyclegroup*} [**,...**]
      [**fdignore** | **fdnext** | **fdprev**]
      [{(**at** *time* [+*n* **day**[**s**]])] |
      (**schedtime** *time* [+*n* **day**[**s**]])}]
  [**,...**]
  [{**at** *time* [***timezone*** | ***tz*** *tzname*] [+*n* **day**[**s**]] |
  schedtime *time* [***timezone*** | ***tz*** *tzname*] [+*n* **day**[**s**]]}]
  [**until** *time* [**timezone** | **tz** *tzname*] [+*n* **day**[**s**]] [**onuntil** *action*]]
  [**deadline** *time* [**timezone** | **tz** *tzname*] [+*n* **day**[**s**]]]
  [**carryforward**]
  [**matching** {**previous** | **sameday** | **relative from** [**+** | **-**] *time* **to** [**+** | **-**] *time* |
    from *time* [+ | -*n* day[s]] **to** *time* [+ *n* day[s]] [**,...**]}]
  [**follows** {[*netagent::*][*workstation*#]*jobstreamname*[**.***jobname* | **@**] [**previous** |
    sameday | **relative from** [**+** | **-**] *time* **to** [**+** | **-**] *time* |

```
      from time [+|-n day[s]] to time [+|-n day[s]]
        ]} ] [,...]] [...]
    [keysched]
    [limit joblimit]
    [needs { [n] [workstation#]resourcename } [,...] ] [...]
    [opens { [workstation#]"filename" [ (qualifier) ] [,...] }] [...]
    [priority number | hi | go]
    [prompt {promptname|"[:|!]text"} [,...] ] [...]
  :
job-statement
    # comment
    [{at time [timezone|tz tzname] [+n day[s]] |
    schedtime time [timezone|tz tzname] [+n day[s]]}][,...]
    [until time [timezone|tz tzname] [+n day[s]] [onuntil action]
    [deadline time [timezone|tz tzname] [+n day[s]]] [,...]
    [maxdur time | percentage % onmaxdur action]
    [mindur time | percentage % onmindur action]
    [every rate]
    [follows {[netagent::][workstation#]jobstreamname{.jobname @} [previous|
      sameday|relative from [+|-] time to [+|-] time |
      from time [+|-n day[s]] to time [+|-n day[s]]
        ]} ] [,...]] [...]
    [confirmed]
    [critical]
    [keyjob]
    [needs { [n] [workstation#]resourcename } [,...] ] [...]
    [opens { [workstation#]"filename" [ (qualifier) ] [,...] }] [...]
    [priority number | hi | go]
    [prompt {promptname|"[:|!]text"} [,...] ] [...]

[job-statement...]
end
```

Argumentos

A Tabela 48 contém uma descrição resumida das palavras-chave de definição do fluxo de tarefas. Uma descrição detalhada de cada palavra-chave de planejamento é fornecida nas subseções a seguir.

Tabela 48. Lista de Palavras-chave de Planejamento

Palavra-chave	Descrição	Página
at	Define a primeira hora em que a execução de um fluxo de tarefa ou uma tarefa pode ser ativada. Quando definida em um ciclo de execução, especifica a primeira hora que uma tarefa ou um fluxo de tarefa pode ser ativado para esse ciclo de execução específico.	"às" na página 220
carryforward	Transportará o fluxo de tarefas se ele não estiver concluído.	"carryforward" na página 221
comentário	Inclui comentários na definição de um fluxo de tarefas ou em uma tarefa contida no fluxo de tarefas.	"comentário" na página 222
confirmed	Especifica que a conclusão deste job requer confirmação.	"confirmed" na página 222
critical	Especifica que a tarefa é essencial e deve ser gerenciada de forma adequada.	"critical" na página 222
prazo final	Especifica o tempo no qual uma tarefa ou fluxo de tarefas deve ser concluído. Quando definido em um ciclo, especifica o tempo dentro do qual uma tarefa ou um fluxo de tarefa deve ser concluído nesse específico ciclo de execução.	"prazo final" na página 223
description	Contém uma descrição do fluxo de tarefa. O comprimento máximo desse campo é de 120 caracteres.	"descrição" na página 224
draft	Especifica que o processo de geração de plano deve ignorar esse fluxo de tarefas.	"draft" na página 224
end	Marca o final de um fluxo de tarefas.	"end" na página 225

Tabela 48. Lista de Palavras-chave de Planejamento (continuação)

Palavra-chave	Descrição	Página
every	Lança o job repetidamente a uma taxa especificada.	"a cada" na página 225
except	Especifica datas que são exceções para as datas **on** em que o fluxo de tarefa está selecionado para execução.	"except" na página 228
fdignore | fdnext | fdprev	Especifica uma regra que deve ser aplicada quando a data selecionada para exclusão cai em um dia livre.	"except" na página 228
follows	Especifica tarefas e fluxos de tarefa que devem ser concluídos com êxito antes da tarefa ou o fluxo de tarefa que está sendo definido ser ativado.	"follows" na página 232
dias livres	Especifica um calendário de dias livres para calcular os *dias úteis* para o fluxo de tarefas. Ele também pode definir sábados e domingos como *dias úteis*.	"dias livres" na página 235
job statement	Define um job e suas dependências.	"job statement" na página 236
keyjob	Marca uma tarefa como chave no banco de dados e no plano para monitoramento por aplicativos, como o IBM Tivoli Business Systems Manager ou o IBM Tivoli Enterprise Console.	"keyjob" na página 238
keysched	Marca um fluxo de tarefas como chave no banco de dados e no plano para monitoramento por aplicativos, como o IBM Tivoli Business Systems Manager ou o IBM Tivoli Enterprise Console.	"keysched" na página 238
limite	Define um limite no número de tarefas que podem ser lançadas simultaneamente a partir do fluxo de tarefas.	"limite" na página 238
matching	Define os critérios de correspondência utilizados quando um critério de correspondência não é especificado nas especificações follows na definição do fluxo de tarefas ou na definição de tarefas no fluxo de tarefas.	"matching" na página 239
maxdur	Especifica a quantidade máxima de tempo que uma tarefa pode ser executada. É possível expressar esse tempo em minutos, ou como uma porcentagem da duração estimada mais recente da tarefa.	"maxdur" na página 240
mindur	Especifica a menor quantidade de tempo na qual uma tarefa normalmente é executada e concluída.	"mindur" na página 241
needs	Define o número de unidades de um recurso requerido pela tarefa ou fluxo de tarefas antes de poder ser lançado. O número mais alto de recursos do qual o fluxo de tarefas pode ser dependente é 1024.	"needs" na página 243
em	Define as datas em que o fluxo de tarefas é selecionado para execução.	"em" na página 244
opens	Define os arquivos que devem estar acessíveis antes da tarefa ou fluxo de tarefas ser lançado.	"opens" na página 249
onuntil	Especifica a ação a ser realizada em uma tarefa ou fluxo de tarefa cuja hora until foi atingida.	"until" na página 256
priority	Define a prioridade de uma tarefa ou fluxo de tarefas.	"prioridade" na página 251
prompt	Define os prompts que devem ser respondidos antes da tarefa ou fluxo de tarefas ser lançado.	"prompt" na página 252
runcycle	Especifica uma etiqueta com um nome de fácil compreensão para o ciclo de execução.	• "except" na página 228 • "em" na página 244
schedule	Atribui um nome ao fluxo de tarefas.	"planejamento" na página 254
schedtime	Especifica o tempo utilizado para configurar o fluxo de tarefas na linha de tempo do plano para determinar sucessores e predecessores.	"schedtime" na página 253
timezone | tz	Especifica o fuso horário a ser utilizado ao calcular a hora de início.	"timezone" na página 256
until	Define a última hora que uma tarefa ou um fluxo de tarefa pode ser ativado. Quando definido em um ciclo de execução especifica a última hora que uma tarefa ou um fluxo de tarefa pode ser ativado para o ciclo de execução específico.	"until" na página 256
validfrom	Define a data a partir da qual a instância do fluxo de tarefa é iniciado.	"validfrom/validto" na página 258
validto	Indica a data na qual a instância do fluxo de tarefas termina.	"validfrom/validto" na página 258
vartable	Define a tabela de variáveis a ser utilizada pelo fluxo de tarefas e pelo ciclo de execução.	"vartable" na página 259

Nota:

1. Os fluxos de tarefas planejados para execução em estações de trabalho marcados como *ignored* não são incluídos no plano de produção quando o plano é criado ou estendido.
2. Palavras-chave incorretamente digitadas utilizadas em definições de tarefa levam a definições de tarefa truncada armazenadas no banco de dados. De fato,

a palavra-chave incorreta é considerada estranha para a definição de tarefa e, portanto, é interpretada como o nome de tarefa de uma definição de tarefa adicional. Geralmente, essa interpretação incorreta também causa um erro de sintaxe ou um erro de definição de tarefa inexistente para a definição de tarefa adicional.

Regras de Especificação de Fuso Horário

É possível especificar um fuso horário em vários níveis de palavra-chave em uma definição de fluxo de tarefas; isto é:

- Para todo o fluxo de tarefas (inclusive de todas as suas especificações de palavras-chave)
- No nível de restrição de tempo (com as palavras-chave `at`, `deadline`, `schedtime` e `until`)
- Para cada instrução de tarefa incluída

As seguintes regras se aplicam ao resolver os fusos horários especificados em uma definição de fluxo de tarefas:

- Quando você especificar o fuso horário no nível do fluxo de tarefas, isto se aplicará às definições de tempo do ciclo de execução (definido com a palavra-chave `on`), bem como às definições nas restrições de tempo.
- Se você especificar um fuso horário tanto no nível de fluxo de tarefas quanto no nível de restrição de tempo, eles deverão ser os mesmos. Se você não especificar nenhum fuso horário, seja no nível de fluxo de tarefas seja no nível de restrição de tempo, o fuso horário especificado na estação de trabalho será utilizado.
- O fuso horário especificado no nível de tarefa pode diferir daquele especificado no nível de fluxo de tarefas e substituí-lo. Se você não especificar nenhum fuso horário, seja no nível de fluxo de tarefas seja no nível de tarefa, o fuso horário especificado na estação de trabalho que está executando a tarefa será utilizado.

Regras de Especificação de Restrição de Tempo

Em uma definição de fluxo de tarefas, é possível especificar restrições de tempo (com as palavras-chave `at`, `deadline`, `schedtime` e `until`) nos níveis de fluxo de tarefas e de ciclo de execução. Quando os dois níveis são especificados, as restrições de tempo especificadas no nível de ciclo de execução substituem aquelas especificadas no nível de fluxo de tarefas.

Exemplos

Este é um exemplo de uma definição de fluxo de tarefas:

```
SCHEDULE M235062_99#SCHED_FIRST1 VALIDFROM 06/30/2005
ON RUNCYCLE SCHED1_PREDSIMPLE VALIDFROM 07/18/2005 "FREQ=DAILY;INTERVAL=1"
   ( AT 1010 )
ON RUNCYCLE SCHED1_PRED_SIMPLE VALIDFROM 07/18/2005 "FREQ=DAILY;INTERVAL=1"
CARRYFORWARD
PROMPT "parto o no?"
PRIORITY 55
:
M235062_99#JOBMDM
 PRIORITY 30
 NEEDS 16 M235062_99#JOBSLOTS
 PROMPT PRMT3

B236153_00#JOB_FTA
 FOLLOWS JOBMDM
END
```

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

o Guia do Usuário do Console do Dynamic Workload, seção Criando Definições de Fluxo de Tarefas.

Detalhes da Palavra-chave de Definição do Fluxo de Tarefas

Esta seção descreve as palavras-chave de definição do fluxo de tarefas listadas na tabela Tabela 48 na página 217.

às

Especifica a dependência de hora. Se a palavra-chave **at** for utilizada, a tarefa ou o fluxo de tarefas não poderá iniciar antes da hora configurada nessa palavra-chave.

Sintaxe

at *time* [**timezone** | **tz** *tzname*][+*n* **day**[**s**]] [**absolute** | **abs**]

Argumentos

time Especifica um período do dia. Os valores possíveis variam de **0000** a **2359**.

tzname Especifica o fuso horário a ser utilizado ao calcular a hora de início. Consulte Capítulo 16, "Gerenciando fusos horários", na página 597 para obter os nomes dos fusos horários. O padrão é o fuso horário da estação de trabalho em que a tarefa ou fluxo de tarefas é lançado.

Nota: Se uma hora **at** e uma hora **until** ou **deadline** forem especificadas, os fuso horários devem ser iguais.

n Especifica um deslocamento em dias da data e hora de início planejadas.

absolute
Especifica se a data de início é baseada no dia do calendário, e não no dia de produção.

Comentários

Se uma hora **at** não for especificada para uma tarefa ou um fluxo de tarefas, a hora de ativação é determinada por suas dependências e prioridade e sua posição no plano de pré-produção é determinada pelo valor designado para a palavra-chave **schedtime**. Para obter informações adicionais sobre a palavra-chave **schedtime**, consulte "schedtime" na página 253.

Se os horários de início do ciclo de execução e do fluxo de tarefas forem definidos, o horário de início do ciclo de execução terá precedência quando o fluxo de tarefas for planejado com **JNextPlan**. Quando o fluxo de tarefas for ativado com o comando **submit**, o horário de início do ciclo de execução não será utilizado.

O valor de tempo na opção **at** é considerado da seguinte forma:

- Se o valor de hora for inferior ao valor configurado na opção global ***startOfDay***, é considerado como para o dia seguinte.
- Se o valor de hora for superior ao valor configurado na opção global ***startOfDay***, é considerado como para o dia atual.

Se o gerenciador de domínio principal da rede executar com as opções `enLegacyStartOfDayEvaluation` e `enTimeZone` configuradas para`yes` para converter o horário `startOfDay` configurado no gerenciador de domínio principal para o fuso

horário local configurado em cada estação de trabalho na rede, você deverá incluir a palavra-chave **absolute** para que funcione quando uma tarefa ou um fluxo de tarefas for submetido.

Se as palavras-chave **at** e **schedtime** não forem especificadas na definição do fluxo de tarefas, por padrão, a instância da tarefa ou do fluxo de tarefas será posicionada no plano na hora especificada na opção global *startOfDay*.

Exemplos

Os exemplos a seguir assumem que o dia de processamento do Tivoli Workload Scheduler começa às 6h.

- O fluxo de tarefas a seguir, selecionado às terças-feiras, não é ativado antes das 3h da manhã de quarta-feira. Seus dois jobs são lançados o mais breve possível após essa hora.

```
schedule sked7 on tu at 0300:
job1
job2
end
```

- O exemplo a seguir ativa o fluxo de tarefas mysked aos domingos às 8h. As tarefas job1, job2 e job3 são todas ativadas aos domingos.

```
schedule mysked on fr at 0800 + 2 days
:
job1
job2 at 0900
job3 follows job2 at 1200
end
```

- O fuso horário da estação de trabalho `sfran` está definido como America/Los_Angeles e o fuso horário da estação de trabalho `nycity` está definido como America/New_York. O fluxo de tarefas a seguir é selecionado para execução às sextas-feiras. Ele é ativado na estação de trabalho `sfran` às 10h do sábado de America/Los_Angeles. O `job1` é lançado na `sfran` o mais breve possível após essa hora. O `job2` é ativado na `sfran` às 14h de sábado de America/New_York (11h de America/Los_Angeles). O `job3` é ativado na estação de trabalho `nycity` às 16h de sábado de America/New_York (13h de America/Los_Angeles).

```
sfran#schedule sked8 on fr at 1000 + 1 day :
job1
job2 at 1400 tz America/New_York
nycity#job3 at 1600
end
```

carryforward

Torna um fluxo de tarefas elegível para ser transportado para o próximo plano de produção se não for concluído antes do final do plano de produção atual.

Sintaxe

carryforward

Exemplos

O fluxo de tarefa a seguir tem o saldo transportado se suas tarefas não tiverem sido concluídas antes do processamento de pré-produção começar para um novo espaço de tempo de produção.

```
schedule sked43 on th
carryforward
:
job12
job13
job13a
end
```

comentário

Inclui comentários em uma definição de fluxo de tarefas e as tarefas contidas em um fluxo de tarefas.

Sintaxe

*text*

Comentários

Insere uma linha de comentário. O primeiro caractere da linha deve ser um sinal de sustenido #.

É possível incluir comentários em uma definição de fluxo de tarefas imediatamente após a linha com a palavra-chave **schedule** ou em uma tarefa contida em uma definição de fluxo de tarefas imediatamente após a linha *job statement*.

Exemplos

O exemplo a seguir inclui os dois tipos de comentários:

```
schedule wkend on fr at 1830
#########################
# The weekly cleanup jobs
#########################
#
carryforward
:
job1
 # final totals and reports
job2
 # update database
end
```

confirmed

Especifica que a conclusão de uma tarefa deve ser confirmada executando um comando **conman confirm**. Consulte “confirm” na página 372 para obter informações adicionais.

Sintaxe

confirmed

Exemplos

No fluxo de tarefas a seguir, a confirmação da conclusão do `job1` deve ser recebida antes do `job2` e `job3` serem lançados.

```
schedule test1 on fr:
job1 confirmed
job2 follows job1
job3 follows job1
end
```

critical

Especifica que a tarefa é crítica para a missão e deve ser processada de acordo.

Uma tarefa crítica para a missão tem tratamento privilegiado. Considerando seu prazo final e a duração estimada, o planejador:

- Durante a construção do plano ou sempre que ele executa o comando **submit**, calcula a hora de início mais recente em que cada um de seus predecessores pode ser iniciado para que a tarefa cumpra com êxito seu prazo final. Isto é

chamado de *horário de início crítico*. Um horário de início crítico é atribuído à tarefa crítica e a cada um de seus predecessores.

O conjunto completo de predecessores para a tarefa crítica é chamado de *rede crítica* da tarefa.

- Ao executar o plano, recalcula dinamicamente os horários de início críticos dentro da rede crítica.

  Quando um predecessor ameaça comprometer a conclusão oportuna da tarefa crítica, ele é *promovido*; ou seja, utilizando diferentes mecanismos do sistema operacional, como a implementação do comando **nice** no UNIX ou a alteração do nível de prioridade no Windows, ele recebe recursos adicionais e seu envio é priorizado em relação a outras tarefas que estão fora da rede crítica. Esta ação é executada de forma recorrente em qualquer predecessor na rede crítica e, se necessário, na tarefa crítica, uma vez que há um risco dessa tarefa se tornar atrasada.

Importante: Tarefas críticas, incluídas no plano, devido à associação ao ciclo de execução, devem ter um prazo final especificado na tarefa, fluxo de tarefas ou nível do ciclo de execução. Considerando que, as tarefas críticas enviadas no plano por solicitação podem não ter um prazo final especificado, e, neste caso, a opção global `deadlineOffset` é usada.

Sintaxe

critical

prazo final

Especifica o tempo no qual uma tarefa ou fluxo de tarefas deve ser concluído. As tarefas ou os fluxos de tarefas que ainda não foram iniciados ou que estavam em execução quando o prazo final expirou, são considerados *atrasados* no plano. Quando um job (ou fluxo de job) está atrasado, são executadas as seguintes ações:

- A tarefa é mostrada como atrasada em **conman**.
- Um evento é enviado para o Tivoli Enterprise Console e para o IBM Tivoli Business Systems Manager.
- Uma mensagem é emitida para os logs *stdlist* e console.

Quando uma tarefa não é concluída antes de seu prazo final, uma mensagem de aviso é exibida. Se essa tarefa não fizer parte de um fluxo de tarefas transportado e você executar `JnextPlan` enquanto ainda está em execução, a tarefa é inserida em USERJOBS. Nesse caso, outra mensagem de aviso sobre o prazo final expirado é incluída no arquivo *`TWS_home`*`/stdlist/logs/`*`yyyymmdd`*`_TWSMERGE.log`.

Nota: Ao usar a palavra-chave **prazo final**, certifique-se de que a opção **bm check deadline** esteja configurada para um valor maior que 0 no arquivo de configuração `localopts` na estação de trabalho em que você estiver trabalhando. É possível definir a opção **bm check deadline** em cada estação de trabalho na qual você deseja estar ciente sobre a expiração do prazo final, ou, se você desejar obter informações atualizadas sobre o ambiente todo, defina a opção no gerenciador de domínio principal. Prazos finais para tarefas críticas são avaliadas automaticamente, independentemente da opção **bm check deadline**. Para obter informações adicionais sobre a opção **bm check deadline**, consulte o *Guia de Administração*.

Sintaxe

deadline *time* [**timezone** | **tz** *tzname*][+*n***day**[**s**] [**,...**]

Argumentos

time Especifica um período do dia. Os valores possíveis variam de **0000** a **2359**.

tzname Especifica o fuso horário a ser utilizado ao calcular o prazo final. Consulte Capítulo 16, “Gerenciando fusos horários”, na página 597 para obter os nomes dos fusos horários. O padrão é o fuso horário da estação de trabalho em que a tarefa ou fluxo de tarefas é lançado.

n Especifica um deslocamento em dias da hora do prazo final planejada.

Nota: Se uma hora **deadline** e uma hora **until** ou **at** forem especificadas, os fuso horários devem ser os mesmos.

Exemplos

O exemplo a seguir ativa um fluxo de tarefas `sked7` todos os dias e uma tarefa `jobc` para iniciar a execução às 14h30 e ser concluído às 16h.

```
schedule sked7 on everyday :
    jobc at 1430 deadline 1600
end
```

descrição

Inclui uma descrição para o fluxo de tarefas.

Sintaxe

description ”*text*”

Comentários

O comprimento máximo desse campo é de 120 caracteres.

Exemplos

```
schedule test1
description "Revenue at the end of the month"
on monthend
:
job1
job2
job3
end
```

draft

Marca um fluxo de tarefas como rascunho. Um fluxo de tarefas de rascunho não é incluído no plano de pré-produção.

Sintaxe

draft

Comentários

Um fluxo de tarefa de rascunho não é considerado ao determinar dependências e não é incluído no plano de produção. Após remover a palavra-chave draft de um fluxo de tarefas, é necessário executar o comando **JnextPlan** para incluir o fluxo de tarefas no plano de produção e, portanto, no plano de produção.

Exemplos

```
schedule test1 on monthend
draft
:
job1
job2
job3
end
```

end

Marca o final da definição de um fluxo de tarefas.

Sintaxe

end

Exemplos

```
schedule test1 on monthend
:
job1
job2
job3
end << end of job stream >>
```

a cada

Define a taxa de repetição de um job. O job é lançado repetidamente na taxa especificada. Se a tarefa tiver uma dependência que não esteja satisfeita, a iteração será iniciada somente depois que a dependência for satisfeita.

Sintaxe

every *rate*

Argumentos

rate A taxa de repetição expressa em horas e minutos, no formato *hhmm*. A taxa pode ser maior que 24 horas.

Comentários

- A iteração **every** de uma tarefa não parará mesmo se uma das repetições da tarefa for encerrada de forma anormal.
- Se a opção **every** for utilizada sem a dependência **at**, as tarefas reexecutadas serão planejadas respeitando a taxa **every** especificada, começando da hora quando a tarefa realmente iniciou.
- No caso específico que a opção **every** é utilizada com a dependência **at** e uma nova execução é retardada (por uma dependência ou por qualquer outra razão), enquanto o Tivoli Workload Scheduler realinha com a hora **at**, pode haver uma ou duas iterações que não respeitam a taxa **every**. Para todos os outros casos, a taxa every é sempre respeitada.

  O exemplo 2 na página 226 explica como o Tivoli Workload Scheduler realinha com a hora **at** se a tarefa inicia depois da hora **at** definida e algumas iterações são perdidas.
- Se uma instância **every** de uma tarefa não iniciar em seu horário de início esperado, use a opção **bm late every** para configurar o número máximo de minutos que devem decorrer antes do Tivoli Workload Scheduler ignorar a tarefa. O valor da opção deve ser definido no arquivo `<TWSHOME>/localopts`:

  bm late every = xx
  Em que *xx* é o número de minutos.

Esta opção é local para cada agente e, dessa forma, ela deve ser definida em cada agente tolerante a falhas que tiver tarefas **every** com a opção **bm late every** definida.

A opção **bm late every** se aplica apenas a tarefas tanto com a opção **every** quanto a dependência de horário **at** definidas, e ela não causa impacto nas tarefas que possuem apenas a opção **every** definida. Apenas tarefas cuja taxa **every** seja maior que o valor **bm late every** sofreram impacto.

O exemplo 4 na página 227 mostra o comportamento de Tivoli Workload Scheduler quando o atraso de uma instância **every** não excede o valor da opção **bm late every**.

O exemplo 5 na página 228 mostra o comportamento de Tivoli Workload Scheduler quando o atraso de uma instância **every** excede o valor da opção **bm late every**.

O exemplo 6 na página 228 mostra o comportamento de Tivoli Workload Scheduler quando a primeira instância de uma tarefa não executa em seu horário de início esperado e excede o valor da opção **bm late every**.

- Se a palavra-chave **every** estiver definida para uma tarefa quando o Horário de Verão (DST) for desativado, isto é, o relógio é atrasado em uma hora, o comando **every job** estará ciente do DST e executará também durante o segundo intervalo de tempo repetido.

Exemplos

1. O exemplo a seguir executa a tarefa `testjob` a cada hora:

   ```
   testjob every 100
   ```

2. O exemplo a seguir mostra a tarefa `testjob1` definida para execução a cada 15 minutos, entre às 18h e 20h:

   ```
   testjob1 at 1800 every 15 until 2000
   ```

   A tarefa deve ser executada às 18h, 18h15, 18h30 e assim por diante, a cada 15 minutos.

   Se a tarefa for submetida `adhoc` às 18h33, as reexecuções ocorrerão às 18h33, 18h34, 18h45, etc. A razão disso é explicada a seguir.

   Primeiro, observe que em uma tarefa, há dois valores de hora a considerar:

   - O *start_time*; que é a hora quando se espera que a tarefa seja executada. Esse valor está configurado para a hora **at** especificada para a tarefa ou para a hora quando a nova execução deve ser ativada. Esse valor pode ser visualizado utilizando `conman showjobs` antes do início da iteração da tarefa.
   - O *time_started*; que é a hora quando a tarefa realmente inicia, por exemplo 18h33. Esse valor pode ser visualizado utilizando-se `conman showjobs` após o início da iteração da tarefa.

   Como a tarefa `testjob1` foi enviada `adhoc` às 1833, estas são as informações que você verá imediatamente após o envio:

   com `conman showjobs`

   ```
   TESTJOB1 HOLD 1800
   ```

   no arquivo Symphony

   `start_time=1800` (porque espera-se que a tarefa seja executada **às** 18h)

   `time_started=NULL` (porque a tarefa ainda não foi iniciada)

   Como o `start_time` (1800) é menor do que o horário de início (1833), `testjob1` inicia imediatamente e as informações atualizadas se tornam:

com conman showjobs

```
TESTJOB1 SUCC 1833
```

no arquivo Symphony

`start_time=1800` (porque esperava-se que a tarefa fosse executada **às** 18h)

`time_started=1833` (porque a tarefa foi iniciada às 18h33)

Quando **batchman** calcula a hora para a próxima iteração, utiliza os seguintes dados:

`start_time=1800`

`rate=0015`

`current_time=1833`

Como a próxima hora de iteração (1800+0015=1815) ainda seria anterior ao valor *current_time* (1833), **batchman** identifica a última iteração planejada que não foi executada somando ao *start_time* quantos *every_rate* possível sem exceder o *current_time*

1800 + 0015 + 0015 = **1830** < 1833

e, em seguida, emite o comando para executar essa iteração. Supondo que essa iteração seja executada às 1834, as informações, após o início da tarefa, tornam-se as seguintes:

com conman showjobs

```
TESTJOB1 SUCC 1834
```

no arquivo Symphony

`start_time=1830` (porque esperava-se que essa iteração de tarefa fosse executada **ás** 1830)

`time_started=1834` (porque essa iteração de tarefa foi iniciada às 18h34)

Após essa iteração de tarefa ser concluída, **batchman** calcula novamente a hora que a próxima iteração precisa iniciar utilizando estes valores atualizados:

`start_time=1830`

`rate=0015`

`current_time=1834`

O fato de que a hora da próxima iteração (1830+0015=1845) seja posterior ao valor *current_time* (1834), indica a **batchman** que a iteração é recuperada. A hora de iteração, que começa a partir de 1845, pode ser realinhada com as horas de iteração planejadas configuradas na definição de tarefa pelas palavras-chave **at** e **every**.

3. O exemplo a seguir não inicia a iteração da tarefa `testjob2` até que a tarefa `testjob1` tenha sido concluída com êxito:

```
testjob2 every 15 follows testjob1
```

4. No exemplo a seguir, o atraso de uma instância de uma tarefa **every** não excede o valor da opção **bm late every**:

```
bm late every = 10
JOB AT 1400 EVERY 0030
```

Esta tarefa deveria ser executada às 1400, 1430, 1500 e assim por diante, a cada trinta minutos.

Se o servidor estiver inativo das 1435 às 1605, as instâncias às 1500, 1530 e 1600 não serão executadas. Às 1605, Tivoli Workload Scheduler reinicializa. Quando ele analisa o arquivo Symphony, determina que o melhor horário em potencial para a próxima instância da tarefa **every** é 1600. Tivoli Workload Scheduler

verifica se o melhor horário em potencial (1600) excede o atraso máximo permitido para uma tarefa **every** (10 minutos).

Neste caso, o atraso não excede a opção **bm late every**, assim o Tivoli Workload Scheduler se comporta como de costume e cria a instância da tarefa **every** com horário de início configurado como 1600. As instâncias subsequentes estão configuradas às 1630, 1700 e assim por diante, a cada trinta minutos.

5. No exemplo a seguir, o atraso da instância de uma tarefa **every** excede o valor da opção **bm late every**:

   ```
   bm late every = 10
   JOB AT 1400 EVERY 00030
   ```

   Esta tarefa deveria ser executada às 1400, 1430, 1500 e assim por diante, a cada trinta minutos.

   Se o servidor estiver inativo das 1435 às 1620, as instâncias às 1500, 1530 e 1600 não serão executadas. Às 1620, Tivoli Workload Scheduler reinicializa. Quando ele analisa o arquivo Symphony, determina que o melhor horário em potencial para a próxima instância da tarefa **every** é 1600. Tivoli Workload Scheduler verifica se o melhor horário em potencial (1600) excede o atraso máximo permitido para uma instância de uma tarefa **every** (10 minutos).

   Neste caso, o atraso é maior que a opção **bm late every**, assim o Tivoli Workload Scheduler aplica o novo comportamento, ele não ativa a instância da tarefa **every** às 1600 e cria a instância da tarefa **every** com horário de início configurado como 1630.

6. O exemplo a seguir mostra o comportamento de Tivoli Workload Scheduler quando a primeira instância de uma tarefa não executa em seu horário de início esperado e excede o valor da opção **bm late every**:

   ```
   bm late every = 10
   JOB AT 1400 EVERY 00030
   ```

   Esta tarefa deveria ser executada às 1400, 1430, 1500 e assim por diante, a cada trinta minutos.

   Se o servidor estiver inativo das 1000 às 1415, a primeira instância da tarefa não será executada. Às 1415, Tivoli Workload Scheduler reinicializa. Quando ele analisa o arquivo Symphony, determina que a primeira instância desta tarefa **every** não foi executada. Nesse caso, Tivoli Workload Scheduler ativa a tarefa às 1415.

except

Define as datas que são exceções para as datas **on** de um fluxo de tarefas. Consulte “em” na página 244 para obter informações adicionais.

Sintaxe

except [**runcycle** *name*]
[**validfrom** *date*] [**validto** *date*]
[**description** *"text"*]
{*date* | *day* | *calendar* | *request* | *"icalendar"*}
[**,...**]
[**fdignore** | **fdnext** | **fdprev**][**subset** *subsetname* **AND** | **OR**]

Argumentos

runcycle *name*
Especifica uma etiqueta com um nome amigável para o ciclo de execução especificado nas linhas a seguir.

valid from *date* **... valid to***date*
Delimita o espaço de tempo durante o qual o fluxo de tarefas está ativo, ou

seja, o fluxo de tarefas é incluído no plano de produção. Observe que a data especificada como o valor **valid to** não está incluída no ciclo de execução; portanto, nessa data o fluxo de tarefas não está ativo.

description "*text*"
Contém uma descrição do ciclo de execução.

date Especifica um ciclo de execução que é executado em datas específicas. A sintaxe utilizada para esse tipo é:

yyyymmdd [,yyyymmdd][,...]Por exemplo, para um fluxo de tarefas que é planejado para execução em 25 de maio de 2009 e em 12 de junho de 2009, o valor é:

```
on
20090525,20090612
```

day Especifica um ciclo de execução que é executado em dias específicos. A sintaxe utilizada para esse tipo é:

{mo | tu | we | th | fr | sa | su}Por exemplo, para um fluxo de tarefas que está planejado para execução toda segunda-feira, o valor é:

```
on
mo
```

calendário
As datas especificadas com este nome em um calendário. O nome do calendário pode vir seguido de um deslocamento no seguinte formato:

{**+** | **-**}*n* {**day[s]** | **weekday[s]** | **workday[s]**}

Em que:

n O número de dias, dias da semana (menos sábado e domingo) ou dias úteis.

days Todos os dias da semana.

weekdays
Todos os dias da semana, exceto sábado e domingo.

dias úteis
Todos os dias da semana, exceto para Sábados e Domingos (a menos que seja especificado de outra forma com a palavra-chave **freedays**) e para as datas marcadas em um calendário de dias livres designado ou no calendário **holidays**.

request Seleciona o fluxo de tarefas somente quando solicitado. Isto é utilizado para fluxos de job que são selecionados por nome em vez de data. Para evitar que um fluxo de tarefa planejado seja selecionado para **JnextPlan**, altere sua definição para ON REQUEST.

Nota: Ao tentar executar um fluxo de tarefas que contenha horários "on request", considere que:

- "On request" sempre antecede "at".
- "On request" nunca antecede "on".

icalendar
Representa um padrão utilizado para especificar uma regra recorrente que descreve quando um fluxo de tarefa é executado.

A sintaxe utilizada para o ciclo de execução com tipo *icalendar* é o seguinte:

FREQ={DAYLY | WEEKLY | MONTHLY | YEARLY}

[;INTERVAL=[-]*n*]

[;{BYFREEDAY | BYWORKDAY | BYDAY=*weekday_list* **|**

BYMONTHDAY=*monthday_list***}]**

em que o valor padrão para a palavra-chave **INTERVAL** é 1.

Utilizando *icalendar* é possível especificar que um fluxo de tarefa seja executado:

a cada *n* dias
utilizando o seguinte formato:

FREQ=DAILY[;INTERVAL=*n*]

em que o valor configurado para **valid from** é o primeiro dia das datas resultantes.

Por exemplo, para um fluxo de tarefas que está planejado para execução diariamente, o valor é:

```
FREQ=DAILY
```

Para um fluxo de tarefa que está planejado para execução a cada dois dias, o valor é:

```
FREQ=DAILY;INTERVAL=2
```

todo dia livre ou útil
utilizando o seguinte formato:

FREQ=DAILY[;INTERVAL=*n*]

;BYFREEDAY | BYWORKDAY

Por exemplo, para um fluxo de tarefa que está planejado para executar todo dia livre o valor é:

```
FREQ=DAILY;BYFREEDAY
```

Para um fluxo de tarefa que está planejado para execução a cada dois dias úteis, o valor é:

```
FREQ=DAILY;INTERVAL=2;BYWORKDAY
```

a cada *n* semanas em *weekdays* específicos
utilizando o seguinte formato:

FREQ=WEEKLY[;INTERVAL=*n*]

;BYDAY=*weekday_list*

em que o valor configurado para *weekday_list* é

```
[SU][,MO][,TU][,WE][,TH][,FR][,SA]
```

Por exemplo, para um fluxo de tarefas que está planejado para execução toda semana na sexta-feira e no sábado, o valor é:

```
FREQ=WEEKLY;BYDAY=FR,SA
```

Para um fluxo de tarefa que está planejado para execução a cada três semanas na sexta-feira, o valor é:

```
FREQ=WEEKLY;INTERVAL=3;BYDAY=FR
```

a cada *n* meses em datas específicas do mês
utilizando o seguinte formato:

FREQ=MONTHLY[;INTERVAL=*n*]

;BYMONTHDAY=*monthday_list*

em que o valor configurado para *monthday_list* é representado por uma lista de

```
[+number_of_day_from_beginning_of_month]
[-number_of_day_from_end_of_month]
[number_of_day_of_the_month]
```

Por exemplo, para um fluxo de tarefas que está planejado para execução mensal no 27º dia, o valor é:

```
FREQ=MONTHLY;BYMONTHDAY=27
```

Para um fluxo de tarefa que está planejado para execução a cada seis meses no 15º e no último dia do mês, o valor é:

```
FREQ=MONTHLY;INTERVAL=6;BYMONTHDAY=15,-1
```

a cada *n* **meses em dias específicos de semanas específicas**
utilizando o seguinte formato:

FREQ=MONTHLY[;INTERVAL=*n*]

;BYDAY=day_of_m_week_list

em que o valor configurado para *day_of_m_week_list* é representado por uma lista de

```
[+number_of_week_from_beginning_of_month]
[-number_of_week_from_end_of_month]
[weekday]
```

Por exemplo, para um fluxo de tarefas que está planejado para execução mensal na primeira segunda-feira e na última sexta-feira, o valor é:

```
FREQ=MONTHLY;BYDAY=1MO,-1FR
```

Para um fluxo de tarefa que está planejado para execução a cada seis meses na 2ª sexta-feira, o valor é:

```
FREQ=MONTHLY;INTERVAL=6;BYDAY=2TU
```

a cada *n* **anos**
utilizando o seguinte formato:

FREQ=YEARLY[;INTERVAL=*n*]

em que o valor configurado para **valid from** é o primeiro dia das datas resultantes.

Por exemplo, para um fluxo de tarefas que está planejado para execução anualmente, o valor é:

```
FREQ=YEARLY
```

Para um fluxo de tarefas planejado para execução a cada dois anos, o valor é:

```
FREQ=YEARLY;INTERVAL=2
```

fdignore | fdnext | fdprev
Especifica uma regra que deve ser aplicada quando a data selecionada para exclusão cai em um dia livre. Pode ser um dos seguintes:

fdignore
Não excluir a data.

fdnext Exclua o dia útil mais próximo após o dia livre.

fdprev
Exclua o dia útil mais próximo antes do dia livre.

subset *subsetname*
Especifica o nome do subconjunto. Se não for especificado um nome, SUBSET_1 será usado por padrão.

AND | OR
Por padrão, os ciclos de execução em um subconjunto estão em um

relacionamento **OR** lógico, mas é possível alterá-lo para um **AND** lógico, desde que o resultado do grupo do ciclo de execução seja uma data ou conjunto de dados positivo (Inclusive).

Para obter uma explicação sobre as palavras-chave remanescentes contidas na sintaxe de **except**, consulte "em" na página 244.

Comentários

É possível definir várias ocorrências da palavra-chave **except** para o mesmo fluxo de tarefas. Cada ocorrência equivale a um ciclo de execução ao qual se pode associar uma regra para dia livre.

Várias ocorrências **except** devem ser consecutivas dentro da definição de fluxo de tarefas.

Cada ocorrência da palavra-chave pode conter qualquer um dos valores permitidos pela sintaxe **except**.

Exemplos

O exemplo a seguir seleciona o fluxo de tarefas `testskd2` para ser executado todos os dias da semana, exceto naqueles cujas datas aparecem nos calendários chamados `monthend` e `holidays`:

```
schedule testskd2 on weekdays
except monthend,holidays
```

O seguinte exemplo seleciona o fluxo de tarefas `testskd3` para executar todos os dias da semana, exceto 15 de Maio de 2005 e 23 de Maio de 2005:

```
schedule testskd3 on weekdays
except 05/15/2005,05/23/2005
```

O exemplo a seguir seleciona o fluxo de tarefas `testskd4` para ser executado todos os dias, exceto dois dias da semana (menos sábado e domingo), anteriores a qualquer data que apareça em um calendário chamado `monthend`:

```
schedule testskd4 on everyday
except monthend-2 weekdays
```

Selecione o fluxo de tarefas `sked4` para ser executado às segundas-feiras, terças-feiras e em 2 dias da semana anteriores a cada data listada no calendário `monthend`. Se a data de execução for um dia livre, execute o fluxo de tarefa no dia útil seguinte mais próximo. Não execute o fluxo de tarefas às quartas-feiras.

```
schedule sked4
on mo
on tu, MONTHEND -2 weekdays fdnext
except we
```

Selecione o fluxo de tarefas `testskd2` para ser executado todos os dias da semana, exceto nos dias listados em `monthend`. Se uma data em `monthend` cair em um dia livre, exclua o dia útil mais próximo antes dele. Neste exemplo, os dias livres são sábados, domingos e todas as datas listadas no calendário `holidays` padrão.

```
schedule testskd2
on weekdays
except MONTHEND fdprev
```

follows

Define as outras tarefas e fluxos de tarefas que devem ser concluídos com êxito antes de uma tarefa ou fluxo de tarefas ser lançado.

Comentários

Utilize a seguinte sintaxe para fluxos de job:

[**follows** {[*netagent::*][*workstation*#]*jobstreamname*[**.***jobname* | **@**]

[**previous** | **sameday** | **relative from [+/-]** *time* **to [+/-]** *time* | **from** *time* [+/-*n* day[s]] **to** *time* [+/-*n* day[s]]

Utilize a seguinte sintaxe para jobs:

[**follows** {[*netagent::*][*workstation*#]*jobstreamname*{**.***jobname* | **@**}

[**previous** | **sameday** | **relative from [+/-]** *time* **to [+/-]** *time* | **from** *time* [+/-*n* day[s]] **to** *time* [+/-*n* day[s]]

Argumentos

netagent
O nome do agente de rede em que a dependência de interligação de redes está definida.

estação de trabalho
A estação de trabalho na qual a tarefa ou fluxo de tarefas deve ter concluído a execução. O padrão é a mesma estação de trabalho da tarefa ou fluxo de tarefas dependente.

Se uma *workstation* não for especificada com *netagent*, o padrão é a estação de trabalho à qual o agente de rede está conectado.

jobstreamname
O nome do fluxo de tarefas que deve ter sido concluído. Para uma tarefa, o padrão é o mesmo fluxo de tarefas da tarefa dependente.

time Especifica um período do dia. Os valores possíveis variam de **0000** a **2359**.

jobname
O nome da tarefa que deve ter sido concluída. Um sinal `at` (@) pode ser utilizado para indicar que todas as tarefas no fluxo de tarefas devem ser concluídas com êxito.

Comentários

Os critérios de resolução de dependência definem como o fluxo de tarefas ou uma tarefa referida por uma dependência de seqüência correspondem a um fluxo de tarefas específico ou instância de tarefa no plano. Como o plano permite a inclusão de várias instâncias da mesma tarefa ou fluxo de tarefas, é possível identificar a instância que resolve a dependência follows externa com base nos seguintes critérios de resolução:

Precedente mais Próximo
A tarefa ou a instância do fluxo de tarefas que resolve a dependência é a mais próxima que precede a instância que inclui a dependência.

Mesmo Dia
A tarefa ou a instância do fluxo de tarefas que resolve a dependência é a mais próxima no tempo planejado para iniciar no dia em que a instância que inclui a dependência está planejada para execução.

Dentro de um Intervalo Relativo
A tarefa ou a instância do fluxo de tarefas que resolve a dependência é a mais próxima em um intervalo de tempo de sua escolha, que está definida relativamente no horário de início planejado da instância dependente.

Dentro de um Intervalo Absoluto
A tarefa ou a instância do fluxo de tarefas que resolve a dependência é a mais próxima em um intervalo de tempo de sua escolha. O intervalo de tempo não é relacionado ao horário de início planejado da instância dependente.

Independentemente de quais critérios correspondentes são utilizados, se várias instâncias de fluxos de tarefas do predecessor em potencial existirem no intervalo de tempo especificado, a regra utilizada pelo produto para identificar a instância do predecessor correta será a seguinte:

1. Tivoli Workload Scheduler procura a instância mais próxima que precede o horário de início da tarefa ou do fluxo de tarefas dependente. Se uma instância desse tipo existir, esta será a instância do predecessor.
2. Se não houver instância precedente, o Tivoli Workload Scheduler considerará a instância do predecessor correta como a instância mais próxima iniciada após o horário de início da tarefa ou do fluxo de tarefas dependente.

O planejador classifica dependências follows como *internas* quando são especificadas apenas por seu nome da tarefa dentro do fluxo de tarefas. Ele classifica-as como *externas* quando são especificadas no formato *jobStreamName.workstationName.jobName*.

Quando um fluxo de tarefas inclui uma tarefa com uma dependência de sequência que compartilha o mesmo nome de fluxo de tarefas (por exemplo, o fluxo de tarefas `schedA` inclui uma tarefa denominada `job6` que tem uma dependência de sequência em `schedA.job2`), a dependência é incluída no plano como uma dependência de sequência *externa*. Desde a Versão 8.3, ao contrário das versões anteriores, porque o planejador usa os critérios de correspondência `mesmo dia` para resolver dependências externas, dependências originadas dessa maneira nunca são incluídas da primeira vez que o objeto é enviado.

Para obter informações e exemplos adicionais sobre como as dependências follows externas são resolvidas no plano, consulte “Gerenciando Dependências de Acompanhamentos Externos para Tarefas e Fluxos de Tarefas” na página 61.

Exemplos

O exemplo a seguir especifica que o fluxo de tarefa `skedc` não deve ser ativado até a instância de fluxo de tarefa anterior mais próxima `sked4` na estação de trabalho `site1` ter sido concluída com êxito:

```
schedule skedc on fr follows site1#sked4 previous
```

O exemplo a seguir especifica que o fluxo de tarefa `skedc` não deve ser ativado até a instância de fluxo de tarefa `sked4` na estação de trabalho `site1` que é executada entre 12h00 de 3 dias antes e 15h00 do dia após a conclusão bem-sucedida:

```
schedule skedc on fr follows site1#sked4 from 1200 –3 days to 0300 1 day
```

O exemplo a seguir especifica para não ativar o fluxo de tarefas `skedc` até que o fluxo de tarefas `sked4` na estação de trabalho `site1` e a tarefa `joba` no fluxo de tarefas `sked5` na estação de trabalho `site2` tenham sido concluídos com êxito:

```
schedule skedc on fr
follows site1#sked4,site2#sked5.joba
```

Não ative sked6 até jobx do fluxo de tarefa skedx no agente de rede cluster4 ter sido concluído com êxito:

```
sked6 follows cluster4::site4#skedx.jobx
```

O exemplo a seguir especifica para não ativar o jobd até que o joba no mesmo fluxo de tarefas e o job3 no fluxo de tarefas skeda tenham sido concluídos com êxito:

```
jobd follows joba,skeda.job3
```

dias livres

Utilize **dias livres** para especificar o nome de um calendário de dias livres que lista os dias livres para sua empresa. Se e como um fluxo de tarefa é executado nestes dias específicos é definido em uma regra *freedays* durante a configuração do ciclo de execução. O Tivoli Workload Scheduler utiliza esse calendário como base para calcular os *dias úteis* do fluxo de tarefas.

A palavra-chave afeta somente o planejamento dos fluxos de tarefa para os quais é especificada.

Sintaxe

freedays *Calendar_Name* [**-sa**] [**-su**]

Argumentos

Calendar_Name
: O nome do calendário que deve ser utilizado como o calendário de dias livres para o fluxo de tarefa. Se *Calendar_Name* não estiver no banco de dados, o Tivoli Workload Scheduler emitirá uma mensagem de aviso quando você salvar o fluxo de tarefas. Se *Calendar_Name* não estiver no banco de dados quando o comando **schedulr** for executado, o Tivoli Workload Scheduler emitirá uma mensagem de erro e utilizará o calendário padrão **holidays** no lugar. Não utilize os nomes de dias da semana para nomes de calendário.

-sa
: Sábados são *dias úteis*.

-su
: Domingos são *dias úteis*.

Comentários

Se você especificar um calendário de dias de folga na definição de fluxo de tarefa, então o conceito de *workdays* terá o seguinte valor: *workdays = todos os dias excluindo sábado e domingo (a menos que você tenha especificado -sa ou -su juntamente com dias livres) e excluindo todas as datas de Calendar_Name*

Se você não especificar **freedays** na definição do fluxo de tarefas, então: *workdays = todos os dias excluindo sábado e domingo e todas as datas do calendário de feriados*.

Por padrão, *sábado* e *domingo* são considerados como dias livres, a menos que você especifique o contrário incluindo **-sa**, **-su** ou ambos após *Calendar_Name*.

Exemplos

Selecione o fluxo de tarefa sked2 para execução em 01/01/2005 e em todos os dias úteis desde que eles não estejam listados no calendário de dias livres denominado GERMHOL.

```
schedule sked2
freedays GERMHOL
on 01/01/2005, workdays
```

Selecione o fluxo de tarefas `sked3` para ser executado dois dias úteis anteriores a cada data do calendário `PAYCAL`. Dias úteis são todos os dias de segunda a sábado desde que eles não estejam listados no calendário de dias livres denominado `USAHOL`.

```
schedule sked3
freedays USAHOL -sa
on PAYCAL -2 workdays
```

Selecione o fluxo de tarefas `sked3` nas datas listadas no calendário `APDATES`. Se a data selecionada for um dia livre, não execute o fluxo de tarefa. Neste exemplo, domingos e todas as datas listadas no calendário `GERMHOL` devem ser considerados como dias livres. Todos os dias, de segunda a sábado, exceto as datas listadas em `GERMHOL`, são dias úteis.

```
schedule sked3
freedays GERMHOL -sa
on APDATES fdignore
```

Selecione o fluxo de tarefas `testsked3` para ser executado todos os dias da semana, exceto 15/5/2005 e 23/5/2006. Se 23/5/2006 for um dia livre, não o exclua. Neste exemplo, sábados, domingos e todas as datas listadas em `GERMHOL` devem ser considerados como dias livres. Todos os dias, de segunda à sexta-feira, exceto as datas listadas em `GERMHOL`, são dias úteis.

```
schedule testskd3
freedays GERMHOL
on weekdays
except 5/15/2005 fdignore
except 5/23/2006
```

Selecione o fluxo de tarefas `testsked4` para ser executado todos os dias, exceto dois dias da semana anteriores a cada data listada no calendário `MONTHEND`. Se a data a ser excluída for um dia livre, não o exclua, mas exclua o dia útil seguinte mais próximo. Neste exemplo, dias livres são todas as datas listadas em `USAHOL`, enquanto dias úteis são todos os dias de segunda a domingo que não são listados em `USAHOL`.

```
schedule testskd4
freedays USAHOL -sa -su
on everyday
except MONTHEND -2 weekdays fdnext
```

job statement

As tarefas podem ser definidas no banco de dados de forma independente (ou conforme descrito em “Tarefa” na página 711) ou como parte de fluxos de tarefa. Em qualquer um dos casos, as alterações são feitas no banco de dados e não afetam o plano de produção até o início de um novo plano de produção.

Sintaxe

Para definir uma tarefa como parte de um fluxo de tarefas, utilize a seguinte sintaxe na definição do fluxo de tarefas:

[*workstation*#]*jobname* [**as** *newname*]
{**scriptname** *filename* | **docommand** “*command*”}
streamlogon *username*
[**description** “*description*”]
[**tasktype** *tasktype*]
[**interactive**]
[**rccondsucc** "*Success Condition*"]
[**recovery**

{**stop** | **continue** | **rerun**}
[**after** [*workstation*#]*jobname*]
[**abendprompt** "*text*"]]

Para utilizar uma tarefa já definida no banco de dados na definição do fluxo de tarefas, defina *job statement* utilizando a seguinte sintaxe:

[*workstation***#**]*jobname*
[**as** *newname*]

Argumentos

as O nome que você deseja utilizar para referir-se à instância da tarefa nesse fluxo de tarefa.

Para obter outras palavras-chave, consulte "Tarefa" na página 711.

Comentários

Ao definir uma tarefa como parte de um fluxo de tarefa, quando a definição do fluxo de tarefa é incluída no banco de dados, a nova definição de tarefa também é incluída e pode ser referida, a partir de então, também de outros fluxos de tarefa.

Nota: Palavras-chave incorretamente digitadas utilizadas em definições de tarefa levam a definições de tarefa truncada armazenadas no banco de dados. De fato, a palavra-chave incorreta é considerada estranha para a definição de tarefa e, portanto, é interpretada como o nome de tarefa de uma definição de tarefa adicional. Geralmente, essa interpretação incorreta também causa um erro de sintaxe ou um erro de definição de tarefa inexistente para a definição de tarefa adicional.

Quando um fluxo de tarefas é adicionado ou modificado, os atributos ou opções de recuperação de suas tarefas também são adicionados ou modificados. Lembre-se de que, quando você inclui ou substitui um fluxo de tarefas, qualquer modificação à tarefa afetará todos os outros fluxos de tarefas que utilizarem as tarefas. Observe que o relatório de referência cruzada, *xref*, pode ser utilizado para determinar os nomes dos fluxos de tarefa, incluindo uma tarefa específica. Para obter informações adicionais sobre o relatório de referência cruzada, consulte "xref" na página 567.

Nota: Tarefas planejadas para execução nas estações de trabalho marcadas como *ignored* e que pertencem a fluxos de tarefas planejados para execução em estações de trabalho ativas são incluídas no plano apesar de não serem processadas.

Exemplos

O exemplo a seguir define um fluxo de tarefas com três tarefas definidas anteriormente:

```
schedule bkup on fr at 20:00 :
  cpu1#jbk1
  cpu2#jbk2
    needs 1 tape
  cpu3#jbk3
    follows jbk1
end
```

A definição de fluxo de tarefas a seguir contém instruções de tarefa que adicionam ou modificam as definições de duas tarefas no banco de dados:

```
schedule sked4 on mo :
  job1 scriptname "d:\apps\maestro\scripts\jcljob1"
    streamlogon jack
    recovery stop abendprompt "continue production"
```

```
  site1#job2 scriptname "d:\apps\maestro\scripts\jcljob2"
    streamlogon jack
    follows job1
end
```

keyjob

A palavra-chave **keyjob** é utilizada para marcar uma tarefa como chave no banco de dados e no plano e para monitorar por aplicativos, como o Tivoli Business Systems Manager ou o Tivoli Enterprise Console. Consulte o guia *Integração do IBM Tivoli Workload Scheduler com Outros Produtos* para obter informações sobre como ativar o mecanismo do mecanismo de chave.

Sintaxe

keyjob

Exemplos

O seguinte exemplo

```
SCHEDULE cpu1#sched1
ON everyday
KEYSCHED
AT 0100
cpu1#myjob1 KEYJOB
END
```

keysched

A palavra-chave **keysched** é utilizada para marcar um fluxo de tarefas como chave no banco de dados e no plano e para monitoramento por aplicativos, como o Tivoli Business Systems Manager. Consulte o guia *Integração do IBM Tivoli Workload Scheduler com Outros Produtos* para obter informações sobre como ativar o mecanismo do mecanismo de chave.

Sintaxe

keysched

Exemplos

O seguinte exemplo:

```
SCHEDULE cpu1#sched1
ON everyday
KEYSCHED
AT 0100
cpu1#myjob1 KEYJOB
END
```

limite

A palavra-chave **limit** limita o número de tarefas que podem ser executadas simultaneamente em um fluxo de tarefas na mesma CPU.

Sintaxe

limit *joblimit*

Argumentos

joblimit

Especifica o número de tarefas que podem ser executadas ao mesmo tempo no fluxo de tarefas. Os valores possíveis são **0** a **1024**. Se você especificar **0**, impedirá que todas as tarefas sejam ativadas, incluindo aquela com prioridade configurada como **GO** ou **HI**.

Exemplos

O exemplo a seguir limita a cinco o número de tarefas que podem ser executadas simultaneamente no fluxo de tarefas sked2:

```
schedule sked2 on fr
  limit 5 :
```

matching

Configura um padrão para os critérios de correspondência a serem utilizados em todas as dependências follows nas quais um critério de correspondência não é configurado na definição do fluxo de tarefas ou nas tarefas contidas no fluxo de tarefas.

Sintaxe

matching {**previous** | **sameday** | **relative from [+/-]** *time* **to [+/-]** *time*

Argumentos

Para obter informações sobre a palavra-chave utilizada com **matching**, consulte a palavra-chave "follows" na página 232.

Exemplos

O exemplo a seguir mostra a definição do fluxo de tarefas SCHED2 que:

- Contém uma job1 que pode ser executada somente hoje se tiver sido executada ontem.
- Precisa que a instância do fluxo de tarefas SCHED1 executada no mesmo dia seja concluída antes da execução.

```
SCHEDULE PDIVITA1#SCHED2
ON RUNCYCLE RULE1 "FREQ=DAILY;"
ON RUNCYCLE CALENDAR2 CAL1
MATCHING PREVIOUS
FOLLOWS PDIVITA1#SCHED1.@ SAMEDAY
FOLLOWS PDIVITA1#SCHED2.JOB1
:
PDIVITA1#JOB1

PDIVITA1#JOB2
END
```

Nesta amostra, a dependência follows externa de PDIVITA1#SCHED2.JOB1 herda os critérios de correspondência especificados na palavra-chave **matching**.

Comentários

Observe que se você excluir um fluxo de tarefas e, em seguida, incluí-lo novamente no banco de dados, o fluxo de tarefas obterá outro identificador. Por esse motivo, se o fluxo de tarefas contiver dependências de FOLLOWS com critérios correspondentes PREVIOUS, essas dependências não serão correspondidas quando **JnextPlan** for executado novamente porque elas são dependências entre planos que referenciam um identificador antigo.

No exemplo a seguir, se você excluir o fluxo de tarefas JS01, para assegurar a integridade referencial do banco de dados, FOLLOWS TWS851MASTER#JS01.@ também será excluído da definição de JS02 e do plano de Pré-produção.

Se você excluir o fluxo de tarefas JS03, para assegurar a integridade referencial do banco de dados, FOLLOWS TWS851MASTER#JS03.@ PREVIOUS também será excluído da definição de JS02 e do plano de Pré-produção.

Se você excluir o fluxo de tarefas JS02 e, em seguida, incluí-lo novamente no plano, suas dependências FOLLOWS também serão incluídas novamente. Quando o plano for estendido, a dependência FOLLOWS TWS851MASTER#JS03.@ PREVIOUS do fluxo de tarefas JS02 não corresponderá a instância do fluxo de tarefas JS03 vinda do plano anterior e esta dependência não será incluída.

Na próxima extensão de plano, o processo funcionará novamente.

```
SCHEDULE TWS851MASTER#JS01
ON RUNCYCLE RULE1 "FREQ=DAILY;"
:
TWS851MASTER#J02
END
SCHEDULE TWS851MASTER#JS03
ON RUNCYCLE RULE1 "FREQ=DAILY;"
SCHEDTIME 1000
CARRYFORWARD
:
TWS851MASTER#J03
END
SCHEDULE TWS851MASTER#JS02
ON RUNCYCLE RULE1 "FREQ=DAILY;"
:
TWS851MASTER#J01
FOLLOWS TWS851MASTER#JS01.@
FOLLOWS TWS851MASTER#JS03.@ PREVIOUS
END
```

Para evitar este problema, use o comando composer **replace** porque, neste caso, os identificadores do fluxo de tarefas não são alterados.

maxdur

Especifica a quantidade máxima de tempo que uma tarefa pode ser executada. É possível expressar esse tempo em minutos, ou como uma porcentagem da duração estimada mais recente da tarefa. Se uma tarefa estiver em execução e o tempo de duração máxima tiver sido excedido, ocorrerão as seguintes ações:

- Uma das seguintes ações é acionada: Kill ou Continue.
- A tarefa é mostrada como excedida nos seguintes locais:
  - Ao executar **showjob** a partir da linha de comandos **conman**, MaxDurationExceeded é exibido.
  - No Dynamic Workload Console nas propriedades da tarefa para a tarefa.
  - Uma mensagem informativa é gravada no arquivo *TWS_home*/stdlist/logs/*yyyymmdd*_TWSMERGE.log.

Se esta tarefa ainda estiver em execução quando o JnextPlan for executado, a tarefa será inserida no fluxo de tarefas USERJOBS. A configuração de duração máxima não é mantida para a tarefa no fluxo de tarefas USERJOBS e não será monitorada. Para que o fluxo de tarefas seja transportado e evitar que a tarefa seja movida para o fluxo de tarefas USERJOBS, sinalize o fluxo de tarefas original no qual a configuração de duração máxima foi especificada como um fluxo de tarefas de transporte (configurando a palavra-chave carryforward no fluxo de tarefas), se a opção global enCarryForward estiver configurada como yes, caso contrário, configure a opção global enCarryForward como all.

Sintaxe

maxdur *time* | *percentage* % **onmaxdur** *action*

Argumentos

time Especifica uma quantidade de tempo expressa usando a sintaxe *HHHMM*, em que:

HHH Representa o número de horas e é um número que varia de 000 a 500.

MM Representa o número de minutos e é um número que varia de 00 a 59.

percentage
Especifica a porcentagem da duração estimada mais recente. Pode ser um número que varia de 0 a 1000000.

onmaxdur *action*
Especifica a ação a ser acionada em uma tarefa que ainda está em execução quando a duração máxima especificada para esta tarefa tiver sido excedida. A seguir estão os valores possíveis do parâmetro *action*:

Eliminar
Especifique para parar a tarefa em execução. Os jobs eliminados terminam no estado ABEND. Todos os jobs ou fluxos de job dependentes de um job eliminado não são liberados. Jobs eliminados podem ser executados novamente.

Continuar
Especifica permitir que a tarefa em execução continue sendo executada, mesmo que ela tenha excedido a duração de tempo máxima.

Ao enviar um comando **conman** para configurar ou alterar a ação **onmaxdur**, você também deve especificar a palavra-chave **maxdur** juntamente com o argumento **onmaxdur**.

Exemplos

O exemplo a seguir especifica a continuação de uma tarefa em execução, se ela ainda estiver em execução após uma hora e 20 minutos:

```
MAXDUR 80 ONMAXDUR CONT
```

O exemplo a seguir especifica a eliminação de uma tarefa em execução se a tarefa for executada por mais de uma hora e 20 minutos:

```
MAXDUR 80 ONMAXDUR KILL
```

O exemplo a seguir especifica a continuação de uma tarefa em execução, se a tarefa ainda estiver em execução após ter excedido 120% de sua duração máxima, onde a duração máxima é baseada na duração estimada mais recente:

```
MAXDUR 120 % ONMAXDUR KILL
```

mindur

Especifica a menor quantidade de tempo na qual uma tarefa normalmente é executada e concluída. Se uma tarefa for concluída antes que esta quantidade de tempo mínima seja atingida, as seguintes ações serão executadas:

- Uma das seguintes ações é acionada: Encerrar de Forma Anormal, Confirmar ou Continuar.
- A tarefa será mostrada como não tendo atingido sua duração mínima nos seguintes locais apenas se for concluída com SUCCESS:
  - Ao executar **showjobs**, **showjobs ;props** a partir da linha de comandos **conman**, `MinDurationNotReached` é exibido.
  - No Dynamic Workload Console nas propriedades da tarefa para a tarefa.
  - Uma mensagem informativa é gravada no arquivo *TWS_home*`/stdlist/logs/`*yyyymmdd*`_TWSMERGE.log`.

Se esta tarefa ainda estiver em execução quando o JnextPlan for executado, a tarefa será inserida no fluxo de tarefas USERJOBS. A configuração de duração mínima não será mantida para a tarefa no fluxo de tarefas USERJOBS e não será monitorada. Para que o fluxo de tarefas seja transportado e evitar que a tarefa seja movida para o fluxo de tarefas USERJOBS, sinalize o fluxo de tarefas original no qual a configuração de duração mínima foi especificada como um fluxo de tarefas de transporte (configurando a palavra-chave `carryforward` no fluxo de tarefas), se a opção global `enCarryForward` estiver configurada como `yes`, caso contrário, configure a opção global `enCarryForward` como `all`.

Sintaxe

mindur *time* | *percentage* % **onmindur** *action*

Argumentos

time Especifica uma quantidade de tempo expressa usando a sintaxe *HHHMM*, em que:

HHH Representa o número de horas e é um número que varia de 000 a 500.

MM Representa o número de minutos e é um número que varia de 00 a 59.

percentage
Especifica a porcentagem da duração estimada mais recente. Pode ser um número que varia de 0 a 1000000.

onmindur *action*
Especifica a ação a ser acionada em uma tarefa concluída antes de sua duração mínima. A seguir estão os valores possíveis do parâmetro *action*:

Encerrar de Forma Anormal
A tarefa está configurada como status **ABEND**.

Confirmar
A tarefa está configurada como o status **CONFIRM**. A carga de trabalho requer uma confirmação do usuário para continuar.

Continuar
A carga de trabalho em execução continua sem executar nenhuma ação.

Exemplos

O exemplo a seguir especifica continuar uma carga de trabalho em execução, mesmo que a tarefa não atinja uma duração mínima de pelo menos 80 minutos:

```
MINDUR 120 ONMINDUR CONT
```

O exemplo a seguir especifica configurar o status da tarefa como Error, se a tarefa não for executada por pelo menos 80 minutos:

```
MINDUR 120 ONMINDUR ABEND
```

O exemplo a seguir requer que um usuário confirme a tarefa quando ele atingir 50% ou metade de sua duração mínima estimada mais recente:

```
MINDUR 50 % ONMINDUR CONFIRM
```

needs

A palavra-chave **needs** define os recursos que devem estar disponíveis antes de uma tarefa ou fluxo de tarefas ser ativado. É possível utilizar a palavra-chave **needs** em uma definição do fluxo de tarefas ou na definição das tarefas contidas, não em ambas.

Sintaxe

needs [*n*] [*workstation***#**]*resourcename* [*,...*]

Argumentos

n Especifica o número de unidades de recurso necessárias. Valores possíveis são **1** a **1024** para cada instrução **necessária**. O padrão é 1.

estação de trabalho
Especifica o nome da estação de trabalho na qual o recurso foi definido localmente. Se não for especificada, o padrão é a estação de trabalho na qual a tarefa ou o fluxo de tarefas dependente é executado. Os recursos podem ser utilizados como dependências somente por tarefas e fluxos de tarefas que são executados na estação de trabalho onde o recurso está definido.

Devido ao mecanismo de resolução de dependências de recursos, uma dependência de recurso no nível do fluxo de tarefas pode ser considerado 'local' (e, então, seu uso suportado) em vez de 'global', quando o fluxo de tarefas e todas as suas tarefas forem definidos na mesma estação de trabalho que o recurso.

No entanto, um agente padrão e seu host podem fazer referência aos mesmos recursos.

resourcename
Especifica o nome do recurso.

Comentários

Uma tarefa ou um fluxo de tarefas pode solicitar no máximo 1024 unidades de um recurso em uma instrução **needs**. No tempo de execução, cada instrução **needs** é convertida em *holders* , cada um contendo no máximo 32 unidades de um recurso específico. Independentemente da quantidade de unidades disponíveis do recurso, para um único recurso, pode haver no máximo 32 portadores. Se 32 portadores já estiverem definidos para um recurso, a próxima tarefa ou fluxo de tarefa que está aguardando esse recurso espera até um portador atual ser encerrado E a quantidade necessária de recursos tornar-se disponível.

Exemplos

O exemplo a seguir impede que o fluxo de tarefas `sked3` seja lançado até que três unidades de `cputime` e duas de `tapes` se tornem disponíveis:

```
schedule sked3 on fr
  needs 3 cputime,2 tapes :
```

O recurso `jlimit` foi definido com duas unidades disponíveis. O exemplo a seguir permite que não mais de duas tarefas sejam executadas simultaneamente no fluxo de tarefas `sked4`:

```
schedule sked4 on mo,we,fr :
  joba needs 1 jlimit
  jobb needs 1 jlimit
  jobc needs 2 jlimit <<executado sozinho>>
  jobd needs 1 jlimit
end
```

em

Essa é uma palavra-chave de fluxo de tarefas que define quando e com que freqüência um fluxo de tarefas será selecionado para execução. Se omitido, o fluxo de tarefas não é incluído no plano de pré-produção. A palavra-chave **on** deve vir depois da palavra-chave **schedule**. Consulte "except" na página 228 para obter informações adicionais.

Sintaxe

on [**runcycle** *name*]

[**valid from** *date*] [**valid to** *date*]

[**description** *"text"*]

[**vartable** *table_name*]

{*date* | *day* | *calendar* | *request* | *"icalendar"*} [**,...**]

[**fdignore** | **fdnext** | **fdprev**][**subset** *subsetname* **AND** | **OR**]

Argumentos

runcycle *name*
: Especifica uma etiqueta com um nome amigável para o ciclo de execução especificado nas linhas a seguir.

valid from *date* **... valid to***date*
: Delimita o espaço de tempo durante o qual o fluxo de tarefas está ativo, ou seja, o fluxo de tarefas é incluído no plano de produção. Observe que a data especificada como o valor **valid to** não está incluída no ciclo de execução; portanto, nessa data o fluxo de tarefas não está ativo.

description "*text***"**
: Contém uma descrição do ciclo de execução.

vartable
: Especifica o nome da tabela de variáveis a ser utilizada pelo ciclo de execução.

date
: Especifica um ciclo de execução que é executado em datas específicas. A sintaxe utilizada para esse tipo é:

  yyyymmdd [,yyyymmdd][,...]Por exemplo, para um fluxo de tarefas que é planejado para execução em 25 de maio de 2009 e em 12 de junho de 2009, o valor é:

  ```
  on
  20090525,20090612
  ```

day
: Especifica um ciclo de execução que é executado em dias específicos. A sintaxe utilizada para esse tipo é:

  {mo | tu | we | th | fr | sa | su}Por exemplo, para um fluxo de tarefas que está planejado para execução toda segunda-feira, o valor é:

  ```
  on
  mo
  ```

calendário

As datas especificadas com este nome em um calendário. O nome do calendário pode vir seguido de um deslocamento no seguinte formato:

{**+** | **-**}*n* {**day[s]** | **weekday[s]** | **workday[s]**}

Em que:

n O número de dias, dias da semana (menos sábado e domingo) ou dias úteis.

days Todos os dias da semana.

weekdays
Todos os dias da semana, exceto sábado e domingo.

dias úteis
Todos os dias da semana, exceto para Sábados e Domingos (a menos que seja especificado de outra forma com a palavra-chave **freedays**) e para as datas marcadas em um calendário de dias livres designado ou no calendário **holidays**.

request Seleciona o fluxo de tarefas somente quando solicitado. Isto é utilizado para fluxos de job que são selecionados por nome em vez de data. Para evitar que um fluxo de tarefa planejado seja selecionado para **JnextPlan**, altere sua definição para ON REQUEST.

Nota: Ao tentar executar um fluxo de tarefas que contenha horários "on request", considere que:

- "On request" sempre antecede "at".
- "On request" nunca antecede "on".

icalendar

Representa um padrão utilizado para especificar uma regra recorrente que descreve quando um fluxo de tarefa é executado.

A sintaxe utilizada para o ciclo de execução com tipo *icalendar* é o seguinte:

FREQ={DAYLY | WEEKLY | MONTHLY | YEARLY}

[;INTERVAL=[-]*n*]

[;{BYFREEDAY | BYWORKDAY | BYDAY=*weekday_list* |

BYMONTHDAY=*monthday_list***}]**

em que o valor padrão para a palavra-chave **INTERVAL** é 1.

Utilizando *icalendar* é possível especificar que um fluxo de tarefa seja executado:

a cada *n* **dias**
utilizando o seguinte formato:

FREQ=DAILY[;INTERVAL=*n*]

em que o valor configurado para **valid from** é o primeiro dia das datas resultantes.

Por exemplo, para um fluxo de tarefas que está planejado para execução diariamente, o valor é:

```
FREQ=DAILY
```

Para um fluxo de tarefa que está planejado para execução a cada dois dias, o valor é:

```
FREQ=DAILY;INTERVAL=2
```

todo dia livre ou útil
utilizando o seguinte formato:

FREQ=DAILY[;INTERVAL=*n*]

;BYFREEDAY | BYWORKDAY

Por exemplo, para um fluxo de tarefa que está planejado para executar todo dia livre o valor é:

```
FREQ=DAILY;BYFREEDAY
```

Para um fluxo de tarefa que está planejado para execução a cada dois dias úteis, o valor é:

```
FREQ=DAILY;INTERVAL=2;BYWORKDAY
```

a cada *n* semanas em *weekdays* específicos
utilizando o seguinte formato:

FREQ=WEEKLY[;INTERVAL=*n*]

;BYDAY=*weekday_list*

em que o valor configurado para *weekday_list* é

```
[SU][,MO][,TU][,WE][,TH][,FR][,SA]
```

Por exemplo, para um fluxo de tarefas que está planejado para execução toda semana na sexta-feira e no sábado, o valor é:

```
FREQ=WEEKLY;BYDAY=FR,SA
```

Para um fluxo de tarefa que está planejado para execução a cada três semanas na sexta-feira, o valor é:

```
FREQ=WEEKLY;INTERVAL=3;BYDAY=FR
```

a cada *n* meses em datas específicas do mês
utilizando o seguinte formato:

FREQ=MONTHLY[;INTERVAL=*n*]

;BYMONTHDAY=*monthday_list*

em que o valor configurado para *monthday_list* é representado por uma lista de

```
[+number_of_day_from_beginning_of_month]
[-number_of_day_from_end_of_month]
[number_of_day_of_the_month]
```

Por exemplo, para um fluxo de tarefas que está planejado para execução mensal no 27º dia, o valor é:

```
FREQ=MONTHLY;BYMONTHDAY=27
```

Para um fluxo de tarefa que está planejado para execução a cada seis meses no 15º e no último dia do mês, o valor é:

```
FREQ=MONTHLY;INTERVAL=6;BYMONTHDAY=15,-1
```

a cada *n* meses em dias específicos de semanas específicas
utilizando o seguinte formato:

FREQ=MONTHLY[;INTERVAL=*n*]

;BYDAY=*day_of_m_week_list*

em que o valor configurado para *day_of_m_week_list* é representado por uma lista de

```
[+number_of_week_from_beginning_of_month]
[-number_of_week_from_end_of_month]
[weekday]
```

Por exemplo, para um fluxo de tarefas que está planejado para execução mensal na primeira segunda-feira e na última sexta-feira, o valor é:

```
FREQ=MONTHLY;BYDAY=1MO,-1FR
```

Para um fluxo de tarefa que está planejado para execução a cada seis meses na 2ª sexta-feira, o valor é:

```
FREQ=MONTHLY;INTERVAL=6;BYDAY=2TU
```

a cada *n* **anos**
utilizando o seguinte formato:

FREQ=YEARLY[;INTERVAL=*n*]

em que o valor configurado para **valid from** é o primeiro dia das datas resultantes.

Por exemplo, para um fluxo de tarefas que está planejado para execução anualmente, o valor é:

```
FREQ=YEARLY
```

Para um fluxo de tarefas planejado para execução a cada dois anos, o valor é:

```
FREQ=YEARLY;INTERVAL=2
```

fdignore | fdnext | fdprev
Indica a regra a ser aplicada se a data selecionada para executar a tarefa ou fluxo de tarefa cair em um dia livre. As configurações disponíveis são:

fdignore
Não incluir a data.

fdnext Inclua o dia útil mais próximo após o dia livre.

fdprev
Inclua o dia útil mais próximo antes do dia livre.

[subset *subsetname* **AND | OR]**

subset *subsetname*
Especifica o nome do subconjunto. Se não for especificado um nome, `SUBSET_1` será usado por padrão.

AND | OR
Por padrão, os ciclos de execução em um subconjunto estão em um relacionamento **OR** lógico, mas é possível alterá-lo para um **AND** lógico, desde que o resultado do grupo do ciclo de execução seja uma data ou conjunto de dados positivo (Inclusive).

Comentários

É possível definir várias ocorrências da palavra-chave **on** para o mesmo fluxo de tarefas. Várias ocorrências **on** devem ser consecutivas dentro da definição de fluxo de tarefas. Cada ocorrência equivale a um ciclo de execução ao qual se pode associar uma regra para dia livre.

Cada ocorrência da palavra-chave pode conter qualquer um dos valores permitidos pela sintaxe **on**.

Se os horários de início do ciclo de execução e do fluxo de tarefas forem definidos, o horário de início do ciclo de execução terá precedência quando o fluxo de tarefas for planejado com **`JNextPlan`**. Quando o fluxo de tarefas for ativado com o

comando **submit**, o horário de início do ciclo de execução não será utilizado.

Exemplos

O exemplo a seguir seleciona o fluxo de tarefas sked1 às segundas e quartas-feiras:

```
schedule sked1 on mo,we
```

O exemplo a seguir seleciona o fluxo de tarefas sked3 em 15 de junho de 2008 e nas datas listadas no calendário apdates:

```
schedule sked3 on 6/15/08,apdates
```

O exemplo a seguir seleciona o fluxo de tarefas sked4 em dois dias da semana antes de cada data que aparece no calendário monthend:

```
schedule sked4 on monthend -2 weekdays
```

O exemplo a seguir seleciona o fluxo de tarefas testskd1 todos os dias da semana, exceto às quartas-feiras:

```
schedule testskd1 on weekdays
  except we
```

O exemplo a seguir seleciona o fluxo de tarefas testskd3 todos os dias da semana, exceto em 15 de maio de 2008 e em 24 de maio de 2008:

```
schedule testskd3 on weekdays
  except 05/16/2008,05/24/2008
```

O exemplo a seguir seleciona o fluxo de tarefas testskd4 todos os dias, exceto em dois dias da semana antes de qualquer data que apareça em um calendário chamado monthend:

```
schedule testskd4 on everyday
  except monthend -2 weekdays
```

Selecione o fluxo de tarefas sked1 para ser executado todas as segundas e sextas-feiras e em 29/12/2009. Se segundas-feiras e 29/12/2009 forem dias livres, execute o fluxo de tarefa no dia útil seguinte mais próximo. Se sextas-feiras forem dias livres, execute o fluxo de tarefa no dia precedente mais próximo. Neste exemplo, os dias livres são sábados, domingos e todas as datas listadas no calendário HOLIDAYS padrão. Os dias úteis são todos os dias, de segunda à sexta-feira, que não estiverem listados no calendário HOLIDAYS.

```
schedule sked1
on mo, 12/29/2009 fdnext
on fr fdprev
```

Este exemplo mostra a saída do comando de exibição do fluxo de tarefa testcli definido para execução em diferentes ciclos de execução na estação de trabalho site2:

```
display js=site2#testcli
```

obtido em formato de 120 colunas, configurando *MAESTROCOLUMNS*=120 antes de acessar a linha de comandos **composer**:

```
JobstreamName Workstation Draft Valid From Valid To UpdatedBy UpdatedOn LockedBy
------------- ----------- ----- ---------- -------- --------- --------- --------
TESTCLI  SITE2  Y  08/25/2008 -  mdmDBE4 08/25/2008 mdmDBE4

SCHEDULE W5#TESTCLI VALID FROM 08/25/2008 TIMEZONE ACT
DESCRIPTION "Job stream with several run cycle settings."
DRAFT
ON RUNCYCLE M5 VALID FROM 08/25/2008
   DESCRIPTION "monthly"
   "FREQ=MONTHLY;INTERVAL=5;BYMONTHDAY=-3,1"
   ( AT 0000 )
ON RUNCYCLE W4 VALID FROM 08/25/2008
   DESCRIPTION "weekly"
   "FREQ=WEEKLY;INTERVAL=5;BYDAY=MO,WE"
   FDNEXT ( AT 0000 )
ON RUNCYCLE D3 VALID FROM 08/25/2008
   DESCRIPTION "daily"
   "FREQ=DAILY;INTERVAL=2"
   FDPREV ( AT 0000 )
```

```
ON RUNCYCLE C2 VALID FROM 08/25/2008
   DESCRIPTION "calendar"
   ITALY +2 DAYS
   ( AT 0000 )
ON RUNCYCLE M6 VALID FROM 08/25/2008
   DESCRIPTION "monthly"
   "FREQ=MONTHLY;INTERVAL=2;BYDAY=1MO,1TH,2WE"
   ( AT 0000 +2 DAYS )
ON RUNCYCLE Y7 VALID FROM 08/25/2008
   DESCRIPTION "yearly"
   "FREQ=YEARLY;INTERVAL=7"
   ( AT 0100 )
ON RUNCYCLE SS1 VALID FROM 08/25/2008
   08/10/2008,08/18/2008,08/20/2008,08/25/2008
   ( AT 0000 UNTIL 0000 +1 DAYS ONUNTIL SUPPR DEADLINE 0000 +2 DAYS )
EXCEPT RUNCYCLE S1 VALID FROM 08/25/2008
   DESCRIPTION "simple"
   08/26/2008,08/28/2008,08/30/2008,09/13/2008
   ( AT 0000 )

CARRYFORWARD
MATCHING SAMEDAY
FOLLOWS LAB235004#SROBY2.@
FOLLOWS X8#COPYOFJS2.RR
FOLLOWS XA15::TPA
KEYSCHED
LIMIT 22
PRIORITY 15
:
X8#PIPPO AS JOBTC
 CONFIRMED
 PRIORITY 13
 KEYJOB
 FOLLOWS W5#POPO.@
 FOLLOWS X8#JS2.F3
END

AWSBIA291I Total objects: 1
```

O calendário ITALY é um calendário customizado definido no banco de dados que configura os dias úteis e os feriados do calendário em utilização na Itália.

opens

Especifica os arquivos que devem estar disponíveis antes de uma tarefa ou fluxo de tarefas ser lançado.

Sintaxe

opens [*workstation#*]"*filename*" [**(***qualifier***)**] [**,**...]

Argumentos

estação de trabalho
: Especifica o nome da estação de trabalho ou classe de estação de trabalho na qual o arquivo existe. O padrão é a estação de trabalho ou classe de estação de trabalho da tarefa ou fluxo de tarefas dependente. Se você utilizar uma classe de estação de trabalho, ela deverá ser a mesma daquela do fluxo de tarefas que inclui esta instrução.

nome do arquivo
: Especifica o nome do arquivo, entre aspas. É possível usar parâmetros do Tivoli Workload Scheduler como parte ou toda a sequência de nomes de arquivos. Se utilizar um parâmetro, ele deverá ser colocado entre acentos circunflexos (^). Consulte “Definição de Variável e Parâmetro” na página 198 para obter informações adicionais e exemplos.

qualifier
: Especifica uma condição de teste válida. No UNIX, o qualificador é transmitido a um comando **test**, que é executado como **root** em bin/sh.

  No Windows, a função de teste é executada como o usuário do Tivoli Workload Scheduler.

  Os qualificadores válidos são:

  -d %p Verdadeiro se o arquivo existir e for um diretório.

-e %p Verdadeiro se o arquivo existir.

-f %p Verdadeiro se o arquivo existir e for um arquivo regular.

-r %p Verdadeiro se o arquivo existir e for legível.

-s %p Verdadeiro se o arquivo existir e seu tamanho for maior que zero.

-w %p Verdadeiro se o arquivo existir e for gravável.

-a Operador booleano AND.

-o Operador booleano OR.

No UNIX e no Windows, a expressão **%p** é utilizada para transmitir o valor designado para *filename* para a função de teste.

Digitar **(notempty)** é o mesmo que digitar **(-s %p)**. Se nenhum qualificador for especificado, o padrão será **-f %p**.

Comentários

A combinação do *caminho do arquivo* e dos *qualificadores* não pode exceder 120 caracteres e o *nome do arquivo* não pode exceder 28 caracteres.

Exemplos

O exemplo a seguir verifica se o arquivo `c:\users\fred\datafiles\file88` na estação de trabalho `nt5` está disponível para acesso à leitura antes de lançar `ux2#sked6`:

```
schedule ux2#sked6 on tu opens nt5#"c:\users\fred\datafiles\file88"
```

O exemplo a seguir verifica se três diretórios, `/john`, `/mary` e `/roger`, existem sob `/users` antes de ativar a tarefa `jobr2`:

```
jobr2 opens "/users"(-d %p/john -a -d %p/mary -a -d %p/roger)
```

O exemplo a seguir verifica se `cron` criou seu arquivo FIFO antes de lançar o job `job6`:

```
job6 opens "/usr/lib/cron/FIFO"(-p %p)
```

O exemplo a seguir verifica se o arquivo `d:\work\john\execit1` na estação de trabalho `dev3` existe e não está vazio, antes de executar o job `jobt2`:

```
jobt2 opens dev3#"d:\work\john\execit1"(notempty)
```

O exemplo a seguir verifica se o arquivo `c:\tech\checker\startf` na estação de trabalho `nyc` existe, se não está vazio e se é gravável, antes de executar o job `job77`:

```
job77 opens nyc#"C:\tech\checker\startf"(-s %p -a -w %p)
```

Segurança para Comandos test(1):
No UNIX, um recurso de segurança especial evita a utilização não autorizado de outros comandos no qualificador. Por exemplo, o arquivo a seguir contém um comando no qualificador:

```
/users/xpr/hp3000/send2(-n "`ls /users/xpr/hp3000/m*`" -o -r %p)
```

Se o qualificador contiver outro comando, as seguintes verificações serão feitas:

- A opção local ***jm no root*** deve ser configurada para no.
- No arquivo de segurança, o usuário que documenta o planejamento ou que inclui a dependência Open Files com um comando **conman adddep**, deve ter acesso submit a um job com os seguintes atributos:
  name=`cmdstest.fileeq`

logon=root
jcl=o caminho dos arquivos abertos
cpu=a CPU na qual os arquivos abertos residem

Observe que `cmdstest` e `fileeq` não existem.

prioridade

Define a prioridade de uma tarefa ou fluxo de tarefas. A designação de uma prioridade diferente às tarefas ou fluxos de tarefas permite determinar qual será iniciado primeiro, se as dependências forem resolvidas.

Assumindo que as tarefas e os fluxos de tarefas estejam prontos para ativação, se você definir uma prioridade para os fluxos de tarefas e para as tarefas nos fluxos de tarefas:

- O fluxo de tarefas que for iniciado primeiro será aquele com a prioridade mais alta.
- Entre as tarefas no fluxo de tarefas com a prioridade mais alta, a tarefa que for iniciada primeiro será aquela com a prioridade mais alta.

Sintaxe

priority *number* | **hi** | **go**

Argumentos

número
Especifica a prioridade. Os valores possíveis são de **0** a **99**. Uma prioridade 0 evita que a tarefa ou o fluxo de tarefas seja ativado.

hi Representa um valor maior do que qualquer valor que possa ser especificado com um número. Quando configurado, a tarefa ou fluxo de tarefa é ativado imediatamente assim que ele esteja livre de todas as dependências.

ir Representa a prioridade mais alta que pode ser configurada. Quando configurado, a tarefa ou fluxo de tarefa é ativado imediatamente assim que ele esteja livre de todas as dependências.

Comentários

Tarefas e fluxos de tarefa com níveis de prioridade **hi** ou **go** são ativados assim que todas as suas dependências sejam resolvidas. Neste caso:

- Os fluxos de tarefa sunstituem o limite de tarefa da cpu.
- As tarefas substituem o limite de tarefa de cpu, mas elas não substituem o limite de tarefas planejadas nem o fence de tarefa de cpu.

Exemplos

O exemplo a seguir mostra o relacionamento entre as prioridades do fluxo de tarefas e as prioridades de tarefa. Os dois fluxos de tarefa, `sked1` e `sked2`, têm as seguintes definições no banco de dados:

```
schedule sked1 on tu
priority 50
:
job1 priority 15
job2 priority 10
end
```

```
schedule sked2 on tu
priority 10
:
joba priority 60
jobb priority 50
end
```

Como o fluxo de tarefas `sked1` tem a prioridade mais alta, então, as tarefas são ativadas na seguinte ordem: `job1`, `job2`, `joba`, `jobb`.

Se, em vez disso, as prioridades do fluxo de tarefas forem as mesmas, as tarefas serão ativadas na seguinte ordem: `joba`, `jobb`, `job1`, `job2`.

Se o `job2` tiver uma dependência **A** e o `job1` tiver uma dependência **B** e a dependência **A** for resolvida (enquanto que a **B** permanece sem solução), o `job2` será iniciado antes de `job1`, embora o `job2` tenha uma prioridade inferior do que a definida para o `job1`.

prompt

Especifica os prompts que devem ser respondidos afirmativamente antes de uma tarefa ou fluxo de tarefas ser lançado.

Sintaxe

prompt *promptname* [**,**...]

prompt "[**:** | **!**]*text*" [**,**...]

Argumentos

promptname
: Especifica o nome de um prompt no banco de dados. É possível especificar mais de um *promptname* separado por vírgulas, mas não pode misturar sob a mesma palavra-chave **prompt** prompts definidos no banco de dados com prompts literais.

text
: Especifica um prompt literal como uma cadeia de texto entre aspas ("). Múltiplas cadeias separadas por barra invertida **n** (\n) podem ser utilizadas para mensagens longas. Se a cadeia começar com dois pontos (:), a mensagem será exibida, mas não será necessária nenhuma resposta. Se a cadeia começar com um ponto de exclamação (!), a mensagem será exibida, mas não será gravada no arquivo de log. É possível incluir barra invertida **n** (\n) dentro do texto para obter novas linhas.

  É possível utilizar um ou mais parâmetros como toda a cadeia de texto ou parte dela. Para utilizar um parâmetro, coloque seu nome entre sinais de interpolação (^). Consulte “Definição de Variável e Parâmetro” na página 198 para obter informações adicionais e exemplos.

  Nota: Quando um aviso local não estiver especificando um parâmetro, o acento circunflexo (^) deverá ser precedido por uma barra invertida (\) ou ocorrerão erros no aviso. Em avisos globais, os acentos circunflexos não precisam ser precedidos por uma barra invertida.

Exemplos

O exemplo a seguir mostra os avisos literal e nomeado. O primeiro é um prompt literal que utiliza um parâmetro chamado `sys1`. Quando uma única resposta afirmativa é recebida para o aviso nomeado`apmsg`, as dependências de `job1` e `job2` são atendias.

```
schedule sked3 on tu,th
  prompt "All ap users logged out of ^sys1^? (y/n)"
:
  job1 prompt apmsg
  job2 prompt apmsg
end
```

O exemplo a seguir define um aviso literal que aparecerá em mais de uma linha. Ele é definido com barra invertida **n** (\n) no final de cada linha:

```
schedule sked5 on fr
  prompt "The jobs in this job stream consume \n
an enormous amount of cpu time.\n
Do you want to launch it now? (y/n)"
:
  j1
  j2 follows j1
end
```

schedtime

Representa a hora em que o fluxo de tarefas é posicionado no plano. Um valor designado para **schedtime** não representa uma dependência para o fluxo de tarefas. Enquanto o plano de produção está em progresso, a instância da tarefa ou do fluxo de tarefas pode iniciar processamento antes da hora configurada na palavra-chave **schedtime** se todas as suas dependências forem resolvidas e se sua prioridade permitir que seja iniciada.

Sintaxe

schedtime *time* [**timezone** | **tz** *tzname*][+*n***day**[**s**]] [**,...**]

Argumentos

time Especifica um horário do dia no formato: HHHHmm. Valores possíveis vão de **0000** a **320000**.

tzname Especifica o fuso horário a ser utilizado ao calcular o horário de início. Consulte Capítulo 16, “Gerenciando fusos horários”, na página 597 para obter os nomes dos fusos horários. O padrão é o fuso horário da estação de trabalho em que a tarefa ou fluxo de tarefas é lançado.

n Especifica um deslocamento em dias da data e hora de início planejadas.

Comentários

Diferentemente da chave **at**, a chave **schedtime** não representa uma dependência de hora, ou seja, não indica uma hora antes da qual uma tarefa ou um fluxo de tarefas não pode ser iniciado. Em vez disso, o valor especificado na palavra-chave **schedtime** é utilizado somente para posicionar a tarefa ou instância do fluxo de tarefas específica no plano de pré-produção. Enquanto o plano de produção está em progresso, a instância da tarefa ou do fluxo de tarefas pode iniciar processamento antes da hora configurada na palavra-chave **schedtime** se todas as suas dependências forem resolvidas e se sua prioridade permitir que seja iniciada.

Para obter uma explicação sobre como a palavra-chave **schedtime** é utilizada para identificar predecessores no plano de pré-produção, consulte “Gerenciando Dependências de Acompanhamentos Externos para Tarefas e Fluxos de Tarefas” na página 61.

As palavras-chave **at** e **schedtime** são mutuamente exclusivas. Se **schedtime** não for especificada e a palavra-chave **at** for especificada na tarefa ou fluxo de tarefas, seu valor será utilizado para posicionar a instância no plano de pré-produção.

Se nem a palavra-chave **at** nem **schedtime** forem especificadas na definição da tarefa ou do fluxo de tarefas, supõe-se, por padrão, ser o valor designado para a opção global *startOfDay* configurada no gerenciador de domínio principal.

Para fluxos de tarefas com uma definição **schedtime**, o valor do campo `Horário de Início` exibido na Dynamic Workload Console depende da configuração da opção global `enPreventStart` (que determina se os fluxos de tarefas sem uma dependência `at` podem ser iniciados imediatamente, sem esperarem pelo ciclo de execução especificado no fluxo de tarefas):

- Se `enPreventStart` estiver configurado como `yes`, o horário de início será 00h00 convertido para o fuso horário especificado na interface gráfica com o usuário.
- Se `enPreventStart` estiver configurado como `no`, o campo de horário de início estará vazio.

Exemplos

Os exemplos a seguir assumem que o dia de processamento do Tivoli Workload Scheduler começa às 6h.

- O fluxo de tarefas a seguir, selecionado às Terças-feiras, foi planejado para iniciar às 3h na manhã de Quarta-feira. Suas duas tarefas são ativadas assim que possível após o fluxo de tarefas começar a ser processado.

```
schedule sked7 on tu schedtime 0300:
job1
job2
end
```

- O fuso horário da estação de trabalho `sfran` está definido como America/Los_Angeles e o fuso horário da estação de trabalho `nycity` está definido como America/New_York. O fluxo de tarefas `sked8` foi selecionado para execução na Sexta-feira. Ele foi planejado para iniciar na estação de trabalho`sfran` às 10h no horário de America/Los_Angeles no Sábado (conforme especificado pelo deslocamento`+ 1 dia`). A tarefa `job1` é ativada no `sfran` assim que possível após o início do processamento do fluxo de tarefas. A tarefa `job2` é ativada em `sfran` às 14h do sábado no horário de America/New_York (11h no horário de America/Los_Angeles). O `job3` é ativado na estação de trabalho `nycity` às 16h de sábado de America/New_York (13h de America/Los_Angeles).

```
sfran#schedule sked8 on fr schedtime 1000 + 1 day:
job1
job2 at 1400 tz America/New_York
nycity#job3 at 1600
end
```

planejamento

Especifica o nome do fluxo de tarefas. Com exceção dos comentários, essa deve ser a primeira palavra-chave em um fluxo de tarefas e deve vir seguida da palavra-chave **on**.

Sintaxe

schedule [*workstation***#**]*jstreamname*

[**timezone** | **tz** *tzname*]

Argumentos

estação de trabalho

Especifica o nome da estação de trabalho na qual o fluxo de tarefas é lançado. O padrão é a estação de trabalho na qual o **composer** é executado para incluir o fluxo de tarefas.

jstreamname

Especifica o nome do fluxo de tarefas. O nome deve iniciar com uma letra e pode conter caracteres alfanuméricos, traços e sublinhados. Ele pode conter até 16 caracteres.

timezone | tz *tzname*

Especifica o fuso horário a ser utilizado ao gerenciar o fluxo de tarefas. Essa configuração é ignorada se a opção global *enTimeZone* estiver configurada para **no** no gerenciador de domínio principal. Para obter informações sobre as configurações do fuso horário, consulte Capítulo 16, “Gerenciando fusos horários”, na página 597.

Comentários

Em uma definição de fluxo de tarefas, é possível configurar um fuso horário para todo o fluxo de tarefas, utilizando a palavra-chave **timezone** no intervalo de validade ou ao especificar as restrições de hora, utilizando **at**, **until** ou **deadline**.

Você também pode configurar um fuso horário para uma tarefa contida em um fluxo de tarefas, configurando as palavras-chave **at**, **until** ou **deadline** para essa tarefa.

Independente de se você está definindo uma tarefa ou um fluxo de tarefas, se você utilizar um fuso horário em uma restrição de hora, por exemplo, **at**, então, deve utilizar o mesmo fuso horário ao especificar as outras restrições de hora, como **deadline** e **until**.

Em uma definição de fluxo de tarefas, é possível configurar um fuso horário para todo o fluxo de tarefas e para as tarefas contidas. Esses fuso horários podem diferir uns dos outros, nesse caso, o fuso horário configurado para a tarefa é convertido no fuso horário configurado para o fluxo de tarefas.

Para gerenciar todas as configurações possíveis de fuso horário, a conversão do fuso horário que é executada ao processar as tarefas e ou fluxos de tarefas na rede do Tivoli Workload Scheduler é realizada respeitando os seguintes critérios:

1. Se um fuso horário não for configurado para uma tarefa de um fluxo de tarefa, então, essa tarefa herda o fuso horário configurado na estação de trabalho onde a tarefa está planejada para execução.
2. Se um fuso horário não for configurado para um fluxo de tarefas, então, o fuso horário configurado é o configurado na estação de trabalho onde está planejada a execução do fluxo de tarefas.
3. Se nenhum dos fusos horários mencionados for configurado, então, o fuso horário utilizado é o configurado para o gerenciador de domínio principal.

Exemplos

Esta é a definição de um fuso horário da tarefa `sked8` na estação de trabalho `sfran` na qual está configurado o fuso horário `America/New_York`. O fuso horário configurado para `job2` para executar na estação de trabalho `LA` está definido como `America/Los_Angeles`.

```
schedule sfran#sked8
tz America/New_York
on fr at 1000 + 1 day:
job1
LA#job2 at 1400 tz America/Los_Angeles
end
```

timezone

Especifica, no nível de fluxo de tarefas, o fuso horário utilizado para calcular o tempo quando o fluxo de tarefas deve iniciar o processamento.

Sintaxe

timezone | **tz** *tzname*

Argumentos

tzname Especifica o nome do fuso horário. Consulte Capítulo 16, "Gerenciando fusos horários", na página 597 para obter os nomes dos fusos horários.

Comentários

O fuso horário especificado no nível de fluxo de tarefas aplica-se às definições de tempo para os ciclos de execução e as restrições de tempo (definidas pelas palavras-chave `at`, `deadline`, `schedtime` e `until`).

Se você especificar um fuso horário para o fluxo de tarefas e um para uma palavra-chave de restrição de tempo, eles deverão ser os mesmos.

Se você não especificar nenhum fuso horário, seja no nível de fluxo de tarefas seja no nível de restrição de tempo, o fuso horário especificado para a estação de trabalho será utilizado.

until

Dependendo da definição de objeto, a palavra-chave until pertence a, especifica a última hora em que um fluxo de tarefas deve ser concluído ou a última hora em que uma tarefa pode ser ativada.

Sintaxe

until *time* [**timezone** | **tz** *tzname*][+*n***day**[**s**]][**absolute** | **abs**][**onuntil** *action*]

Argumentos

time Especifica o período do dia. Os valores possíveis vão de **0000** a **2359**.

tzname Especifica o fuso horário a ser utilizado ao calcular a hora. Consulte Capítulo 16, "Gerenciando fusos horários", na página 597 para obter os nomes dos fusos horários. O padrão é o fuso horário da estação de trabalho em que a tarefa ou fluxo de tarefas é lançado.

Nota: Se uma hora **until** e uma hora **at** ou **deadline** forem especificadas, os fuso horários devem ser iguais.

n Especifica um deslocamento, em dias, da data e hora planejadas.

absolute
Especifica se a data **until** é baseada no dia do calendário, e não no dia de produção.

onuntil *action*
Dependendo da definição de objeto, a palavra-chave until pertence a, especifica:

- A ação a ser realizada em uma tarefa cuja hora until expirou, mas a tarefa ainda não foi iniciada.
- A ação a ser realizada em um fluxo de tarefas cuja hora until expirou, mas o fluxo de tarefas ainda não foi concluído no estado SUCC.

A seguir estão os valores possíveis do parâmetro *action*:

suppr A tarefa ou fluxo de tarefas e todas as tarefas ou fluxos de tarefas dependentes não são executados. Esse é o comportamento padrão.

Depois que a hora until expira em um fluxo de tarefas, o status do fluxo de tarefas é calculado com as regras usuais; as tarefas suprimidas não são consideradas no cálculo. No caso do fluxo de tarefas conter pelo menos uma tarefa **every**, seu status será HOLD.

Quando a hora until de uma tarefa expirar, a tarefa será movida para o status HOLD ou manterá qualquer status anterior que for um status final.

Se a hora **until** for transmitida juntamente com as opções **onuntil suppr** e **carryforward**, o fluxo de tarefas será *transferido* por **JnextPlan** apenas se a data **until** for igual à data quando JnextPlan executar. Se as datas de execução **until** e JnextPlan não forem as mesmas, o fluxo de tarefas não será *transferido*.

cont O job ou o fluxo de job é executado quando todas as condições necessárias são atendidas e uma mensagem de notificação é gravada no log quando a hora until decorre.

Se a hora **until** for transmitida juntamente com as opções **onuntil cont** e **carryforward**, o fluxo de tarefas será sempre *transferido* por **JnextPlan**.

canc Uma tarefa ou um fluxo de tarefas é cancelado quando a hora **until** especificada expira. Ao utilizar *onuntil canc* nas tarefas, a operação de cancelamento na tarefa será emitida pelo FTA no qual a tarefa é executada. Todas as tarefas ou fluxos de tarefas dependentes da conclusão de uma tarefa ou um fluxo de tarefa que foi cancelado, serão executados porque a dependência não existe mais.

Se a hora **until** for transmitida juntamente com as opções **onuntil canc** e **carryforward**, o fluxo de tarefas não será *transferido* por **JnextPlan**, pois ele já está cancelado.

Nota: Ao utilizar *onuntil canc* no nível do fluxo de tarefas, defina como proprietário do fluxo de tarefas a estação de trabalho superior na hierarquia do ambiente de planejamento, entre todas as estações de trabalho que possuírem tarefas contidas no fluxo de tarefas.

Nota: As palavras-chave **until** e **deadline** podem ser utilizadas na mesma definição, mas devem ser expressas com a mesma definição de fuso horário.

Exemplos

O exemplo a seguir evita que `sked1` seja ativado depois das 17h às terças-feiras:

```
schedule sked1 on tu until 1700 :
```

O exemplo a seguir ativa `sked1` às 17h, quando sua hora **until** é alcançada:

```
schedule sked1 until 1700 onuntil cont
```

O exemplo a seguir ativa o `job1` entre as 13h e as 17h nos dias de semana:

```
schedule sked2 on weekdays :
  job1 at 1300 until 1700
end
```

O exemplo a seguir ativa o `joba` a cada 15 minutos entre as 22h30 e as 23h30 às segundas-feiras:

```
schedule sked3 on mo :
  joba at 2230 every 0015 until 2330
end
```

O exemplo a seguir ativa o fluxo de tarefas `sked4` aos domingos entre as 8h e as 13h. As tarefas devem ser ativadas dentro deste intervalo:

```
schedule sked4 on fr at 0800 + 2 days
  until 1300 + 2 days
:
  job1
  job2 at 0900 <<lançado no domingo>>
  job3 follows job2 at 1200 <<lançado no domingo>>
end
```

O exemplo a seguir ativa o fluxo de tarefas `sked8` em dias da semana às 16h e a execução deve ser concluída às 17h. Se o fluxo de tarefas não for concluído às 17h, ele será considerado atrasado. Os jobs são ativados da seguinte maneira: o `job1` é executado às 16h ou o mais tardar, às 16h20, e se o `job1` ainda não tiver sido iniciado nessa hora, será gravada uma mensagem de notificação no log e a execução será iniciada. O `job2` é executado às 16h30, ou o mais tardar às 16h50, e se o `job2` não for iniciado nessa hora, ele será cancelado.

```
schedule sked8 on weekdays at 1600 deadline 1700 :
   job1 at 1600 until 1620 onuntil cont
   job2 at 1630 until 1650 onuntil canc
end
```

O exemplo a seguir ativa o fluxo de job `sked01`. Quando o evento **until** ocorre, o fluxo de job `sked02` é executado porque o fluxo de job `sked01` está colocado no estado SUCC. O fluxo de job `sked03`, por sua vez, não é executado porque tinha uma dependência de tempo exata no job `job01` que não foi liberada.

```
SCHEDULE sked01 on everyday:
job01 until 2035 onuntil suppr
end

SCHEDULE sked02 on everyday follows sked01.@
:
job02
end

SCHEDULE sked03 on everyday follows sked01.job01
:
job03
END
```

validfrom/validto

É possível configurar uma hora de validade para um fluxo de tarefas, que é o espaço de tempo no qual o fluxo de tarefas está incluído no plano de pré-produção. A hora de validade é configurada utilizando a chave **validfrom** na definição do fluxo de tarefas.

Sintaxe

validfrom *date*

Argumentos

validfrom *date*

Define a partir de qual data o fluxo de tarefas está ativo, ou seja, ele deve ser incluído em um plano de produção se a duração do plano de produção incluir essa data.

Comentários

Diferentes *versões* do mesmo fluxo de tarefa podem ser definidas criando-se fluxos de tarefa diferentes com o mesmo nome e estação de trabalho, mas tendo diferentes intervalos de validade. O conceito de versões do mesmo fluxo de tarefas compartilhando o mesmo*jobstreamname* e o mesmo*workstationname* é chave ao gerenciar dependência desse fluxo de tarefas. Na verdade, ao definir dependências follows externas em um fluxo de tarefas, você identifica o fluxo de tarefas predecessor, utilizando seu *jobstreamname* e *workstationname*. O fluxo de tarefas identificado como a dependência é aquele cujo intervalo de validade é durante o período em que a dependência está ativa.

Se você alterar o*jobstreamname* ou o*workstationname* em uma versão do fluxo de tarefas, a alteração é propagada em todas as suas versões.

Se você bloquear uma versão do fluxo de tarefas, todas as versões desse fluxo de tarefas são bloqueadas.

Se você alterar o nome de uma tarefa definida em uma versão do fluxo de tarefas, então, o novo nome de tarefa é propagado em todas as versões do fluxo de tarefas. Isso significa que, se você modificar algo diferente do *jobstreamname* ou do *workstationname*, as associações internas e externas do fluxo de tarefas permanecem consistentes.

Ao definir uma versão do fluxo de tarefas, é solicitado somente digitar a data **validfrom**, e a data **validto** é configurada automaticamente para o valor da data **validfrom** da próxima versão. A data **validto** é mostrada ao emitir os comandos **list** e **display** quando*MAESTROCOLUMNS* está configurado para 120. Diferentes versões do mesmo fluxo de tarefas continuam a compartilhar os campos do nome e da estação de trabalho após sua criação. Se você modificar o nome de uma versão ou alterar a estação de trabalho na qual foi definida, a alteração é aplicada em todas as versões desse fluxo de tarefas.

Nota: Se as palavras-chave usadas na definição de fluxo de tarefas forem **validfrom** e **validto**, as palavras-chave de filtragem correspondente usadas na emissão de comandos contra definições de objetos armazenadas no banco de dados são **validfrom** e **validto**. Para obter informações adicionais, consulte Capítulo 9, “Gerenciando Objetos no Banco de Dados - Composer”, na página 277.

A data especificada como o valor **validto** não foi incluída no ciclo de execução, assim o fluxo de tarefas não está ativo nesta data.

vartable

Utilizando as tabelas de variáveis, você designa valores diferentes para a mesma variável e, dessa forma, reutiliza a mesma variável em definições de tarefas e ao definir dependências de aviso e de arquivo.

Sintaxe

vartable *tablename*

Argumentos

vartable *tablename*
O nome da tabela de variáveis. O nome pode conter até 80 caracteres alfanuméricos, incluindo hifens (-) e sublinhados (_), e deve iniciar com uma letra.

Definição de Regra de Evento

Uma regra de evento de planejamento define um conjunto de ações que devem ser executadas na ocorrência de condições de evento específicas. A definição de uma regra de evento correlaciona eventos e aciona ações.

Uma definição de regra de evento é identificada por um nome da regra e por um conjunto de atributos que especificam se a regra está no estado de rascunho ou não, o intervalo de tempo em que ela está ativa, o período de tempo de sua validade e outras informações necessárias para decidir quando as ações são acionadas. Ela inclui informações relacionadas aos eventos específicos (`eventCondition`) que a regra deve detectar e as ações específicas que ela deve acionar em sua (`ação`) de detecção ou tempo limite. As regras complexas podem incluir vários eventos e várias ações.

Para obter uma visão geral das regras de planejamento de eventos, consulte Capítulo 7, "Automação de Carga de Trabalho Orientada pelo Evento em Execução", na página 119.

Sintaxe

Você define regras de evento diretamente em linguagem XML com o uso de qualquer editor XML. Você pode configurar uma variável de ambiente em seu computador para abrir automaticamente um editor XML de sua escolha para trabalhar com definições de regras de eventos. Consulte "O Editor do Composer" na página 278 para obter detalhes. O XML que descreve a regra de evento deve corresponder aos esquemas de linguagem de regra definidos em `EventRules.xsd` e em `FilteringPredicate.xsd`.

Os esquemas de linguagem de regra definidos em `eventRules.xsd` e em `FilteringPredicate.xsd` são utilizados para validar suas definições de regra e, dependendo do editor de XML que você possua, para fornecer ajuda de sintaxe. Os arquivos estão localizados no subdiretório `schemas` do diretório de instalação do Tivoli Workload Scheduler.

A tabela a seguir lista todos os elementos de linguagem usados para definir uma regra de evento. O Tabela 49 explica o significado da notação que segue cada elemento de linguagem. *n* representa um número solto.

Tabela 49. Explicação da Notação que Define o Número de Ocorrências para um Elemento de Linguagem.

Notação	Significado
(0, 1)	0 indica que o elemento de linguagem é opcional. 1 indica que, se codificado, somente 1 ocorrência é permitida.
(0, n)	0 indica que o elemento de linguagem é opcional. n indica que, se codificado, várias ocorrências são permitidas.

Tabela 49. Explicação da Notação que Define o Número de Ocorrências para um Elemento de Linguagem. (continuação)

Notação	Significado
(1, 1)	O primeiro 1 indica que o elemento de linguagem é necessário. O segundo 1 indica que somente 1 ocorrência é permitida.
(1, 2)	1 indica que o elemento de linguagem é necessário. 2 indica que 2 ocorrências são necessárias.
(1, n)	1 indica que o elemento de linguagem é necessário. n indica que várias ocorrências são permitidas.

- eventRule name=" " ruleType=" " isDraft=" " (1, 1)
  - description (0, 1)
  - timeZone (0, 1)
  - validity from=" " to=" " (0, 1)
  - activeTime start=" " end=" " (0, 1)
  - timeInterval amount=" " unit=" " (0, 1)
  - eventCondition eventProvider=" " eventType=" " (1, n)
    - scope (0, 1)
    - filteringPredicate (0, 1)
      - attributeFilter name=" " operator="eq" (0, n)
        - value (1, n)
      - attributeFilter name=" " operator="ne" (0, n)
        - value (1, n)
      - attributeFilter name=" " operator="le" (0, n)
        - value (1, 1)
      - attributeFilter name=" " operator="ge" (0, n)
        - value (1, 1)
      - attributeFilter name=" " operator="range" (0, 1)
        - value (1, 2)
  - correlationAttributes (0, 1)
    - attribute name=" " (1, n)
  - action actionProvider=" " actionType=" " responseType=" " (0, n)
    - description (0, 1)
    - scope (0, 1)
    - parameter name=" "(1, n)
    - value (1, 1)

As definições de regras de eventos são agrupadas nos conjuntos de regras de eventos.

- eventRuleSet (1, 1)
  - eventRule (1, n)

Utilize o elemento de linguagem `eventRuleSet` também se você precisar incluir uma única definição de regra.

Nota: Nenhum dos comentários escritos em XML no formato `<!--text-->` é salvo no banco de dados. Na próxima vez em que você abrir uma definição de regra, os comentários escritos na primeira vez não estarão presentes. Ao invés disso, use o atributo `description` para escrever quaisquer informações adicionais.

Argumentos

As palavras-chave que descrevem uma regra de evento são as tags XML a seguir:

eventRule

O objeto de planejamento que inclui a definição de várias condições de eventos e várias ações de regras além de um conjunto de atributos que definem quando a regra é ativada. Um elemento `eventRule` geralmente inclui:

- Um número de atributos de regras necessários e opcionais
- Uma ou mais condições de eventos
- Uma ou mais ações de regras, embora as regras sem ações também sejam permitidas

Os atributos de regras são:

- Atributos necessários:

  nome O nome da regra de evento. Pode ter até 40 caracteres alfanuméricos de comprimento, deve iniciar com uma letra e não pode conter espaços em branco. Caracteres de sublinhado (`_`) e traço (`-`) são permitidos.

  ruleType
  O tipo de regra é baseado no número de eventos - e em suas correlações - que a regra está definida para detectar. Pode ser um dos seguintes:

  filter A regra é ativada na detecção de um único evento específico.

  sequence
  A regra é ativada quando uma seqüência ordenada de eventos chega em um intervalo de tempo específico.

  set A regra é ativada quando uma seqüência não ordenada de eventos chega em um intervalo de tempo específico.

  Regras do tipo `set` e `sequence` também podem ser ativadas no **timeout**, quando um ou mais eventos chegam mas a seqüência completa não chega dentro do espaço de tempo especificado.

  isDraft
  Especifica se a definição da regra está ativada atualmente. Os valores podem ser **yes** ou **no**. O padrão é **no**.

- Atributos opcionais:

  descrição
  Uma descrição da regra. Pode ter até 120 caracteres.

  Lembre-se de gravar quaisquer caracteres XML especiais que você possa utilizar na notação de XML especial, tal como:
  – `&` para &
  – `>` para >
  – `<` para <
  – `"` para "

  e assim por diante.

  timeZone
  Especifica um fuso horário diferente para a execução da regra. O fuso horário padrão é o fuso horário definido na estação de trabalho na qual o servidor de processamento de eventos reside.

  A aplicação do DST (Horário de Verão) tem impacto no intervalo `activeTime` (descrito a seguir) das regras de evento:
  – No dia em que o DST é ativado (o relógio é adiantado em uma hora), os tempos de operação da regra previstos para a

hora absorvida pela aplicação do DST são adiantados em uma hora. Por exemplo, 2h10 torna-se 3h10.
- No dia em que o DST é desativado (o relógio é atrasado em uma hora), os tempos de operação da regra previstos para a hora duplicada pela aplicação do DST são observados sem o DST.

validity
Especifica o período de validade da regra em termos de:

de *yyyy-mm-dd*
O período de validade inicia à meia-noite (do fuso horário da regra) do dia especificado.

até *yyyy-mm-dd*
O período de validade termina à meia-noite (do fuso horário da regra) do dia especificado.

activeTime
Especifica o espaço de tempo da atividade da regra em cada dia da validade em termos de:

start *hh:mm:ss*
O início do espaço de tempo em que a regra está ativa em horas, minutos e segundos.

end *hh:mm:ss*
O fim do espaço de tempo em que a regra está ativa em horas, minutos e segundos. Se a hora for anterior à de **início às** *hh:mm:ss*, ela se refere ao dia seguinte.

timeInterval
Se aplica às regras que incluem ações de tempo limite. Especifica o intervalo de tempo dentro do qual todos os eventos especificados na regra devem ter sido recebidos antes de uma ação corretiva de tempo limite ser iniciada. O intervalo de tempo se inicia a partir do momento que o primeiro evento especificado na regra é detectado. Especifique o intervalo de tempo em termos de:

quantidade
A duração do intervalo de tempo em unidades de tempo.

unidade
A unidade de tempo em um dos seguintes formatos:
- horas
- segundos
- milissegundos

Este atributo é obrigatório quando há ações de tempo limite na definição de regra de evento.

eventCondition
A condição de evento é formada pelos seguintes atributos:

- Atributos necessários:

eventProvider
Identifica o provedor de monitoramento de eventos que pode capturar um tipo de evento. Os provedores de eventos fornecidos no momento da instalação são:

TWSObjectsMonitor
Monitora o status dos objetos do plano do Tivoli Workload Scheduler. Este provedor de eventos é executado em cada agente Tivoli Workload Scheduler e envia os eventos para o servidor de processamento de eventos.

TWSApplicationMonitor
Monitora os processos, o sistema de arquivos e as caixas de mensagens do Tivoli Workload Scheduler.

FileMonitor
Monitora os eventos que afetam os arquivos.

eventType
Especifica o tipo de evento que deve ser monitorado. Cada evento pode ser referido como um provedor de evento. As tabelas a seguir listam os tipos de eventos por provedor de evento. Para ver as propriedades de cada tipo de evento, acesse Apêndice A, "Evento de Automação e Definições de Ação da Carga de Trabalho Orientada a Evento", na página 651.

O Tabela 50 lista os eventos `TWSObjectsMonitor`.

Tabela 50. Eventos `TWSObjectsMonitor`.

Tipo de evento	Quando um evento é enviado
JobStatusChanged	Quando o status de uma tarefa é alterado.
JobUntil	A hora de início mais recente de uma tarefa que decorreu
JobSubmit	Uma tarefa é enviada
JobCancel	Uma tarefa é cancelada
JobRestart	Uma tarefa foi reiniciada
JobLate	Uma tarefa foi atrasada
JobPromoted	Uma tarefa em uma rede crítica se aproxima do horário de início crítico e ainda não foi iniciada.
JobRiskLevelChanged	• Uma nova tarefa crítica é incluída no plano com o nível de risco configurado para alto ou em potencial. • O nível de risco para a tarefa é configurado para risco alto ou risco em potencial e o nível do risco é alterado. • Uma tarefa crítica com o nível de risco configurado para alto ou em potencial é removida do plano. Um evento não é enviado se a tarefa crítica estiver no fluxo de tarefas chamado `JOBS`.
JobExceededMaximumDuration	Uma tarefa excede o tempo de duração máximo estabelecido para a tarefa
JobDidnotReachMinimumDuration	A tarefa não é executada durante um período suficiente para atingir o tempo de duração mínimo estabelecido para a tarefa.
JobStreamStatusChanged	O status do fluxo de tarefas é alterado.
JobStreamCompleted	Um fluxo de tarefas foi concluído
JobStreamUntil	A hora de início mais recente decorrida de um fluxo de tarefas
JobStreamSubmit	Um fluxo de tarefas é enviado
JobStreamCancel	Um fluxo de tarefas é cancelado
JobStreamLate	Um fluxo de tarefas foi atrasado
WorkstationStatusChanged	Uma estação de trabalho foi iniciada ou interrompida.
ApplicationServerStatusChanged	O WebSphere Application Server foi interrompido ou está reiniciando
ChildWorkstationLinkChanged	A estação de trabalho definida na regra de eventos é vinculada à, ou desvinculada da, sua estação de trabalho pai (a estação de trabalho pai envia o evento)
ParentWorkstationLinkChanged	A estação de trabalho pai é vinculada à, ou desvinculada da, estação de trabalho definida na regra de evento (a estação de trabalho filha envia o evento)
PromptStatusChanged	Um prompt pergunta ou é respondido
ProductAlert	O arquivo do Symphony na estação de trabalho especificada na regra de evento contém um registro corrompido

Nota:

Qualquer mudança executada em uma estação de trabalho referenciada em uma regra não é relatada na regra. Por exemplo, se você modificar, atualizar ou excluir uma estação de trabalho que seja referenciada em uma regra, a regra ignorará a mudança e continuará a considerar a estação de trabalho no estado em que encontrava quando ela foi incluída na regra. Uma regra com tipo de evento **ParentWorkstationLinkChanged** não é aplicável quando a Estação de Trabalho de Filtros está configurada como agente, conjunto, conjunto dinâmico ou mecanismo remoto, e o atributo **ParentWorkstation** está configurado como broker. Para monitorar uma mudança de status do link entre o servidor do broker de carga de trabalho e uma estação de trabalho gerenciada pelo servidor do broker de carga de trabalho, defina uma regra com tipo de evento igual a **ChildWorkstationLinkChanged**.

Uma regra com tipo de evento igual a **ChildWorkstationLinkChanged** funciona apenas quando a estação de trabalho do broker está vinculada, com link removido, parada ou iniciada. Se a mudança no status do link for devido a uma operação de parada ou início na estação de trabalho do agente com os comandos **StartupLwa** e **ShutdownLwa**, nenhuma ação é iniciada. Para monitorar operações de parada ou início nas estações de trabalho do agente, defina uma regra com um tipo de evento igual a **WorkstationStatusChanged**.

O Tabela 51 lista os eventos `TWSApplicationMonitor`.

Tabela 51. Eventos `TWSApplicationMonitor`.

Tipo de evento	Quando um evento é enviado
MessageQueuesFilling	Uma caixa de correio especificada excede o valor total da porcentagem.
TivoliWorkloadSchedulerFileSystemFilling	O sistema de arquivos no qual a instância do Tivoli Workload Scheduler está instalada excede o valor total da porcentagem.
TivoliWorkloadSchedulerProcessNotRunning	Um processo especificado não está em execução.

O Tabela 52 lista os eventos `FileMonitor`.

Tabela 52. Eventos `FileMonitor`.

Tipo de evento	Quando um evento é enviado
FileCreated	Um arquivo é criado
FileDeleted	Um arquivo é excluído
ModificationCompleted	Um arquivo é modificado (o evento será enviado apenas se dois ciclos de monitoramento consecutivos forem transmitidos desde que o arquivo foi criado ou modificado sem nenhuma mudança adicional sendo detectada)
LoggedMessageWritten	Uma sequência específica foi localizada em um arquivo (esse evento pode ser usado para monitorar os logs do aplicativo ou do sistema)

- Atributos opcionais:

  scope Um ou mais atributo de qualificação que descrevem o evento. Ele pode ter até 120 caracteres. O escopo é gerado automaticamente a partir do que é definido no XML. Ele não pode ser especificado pelos usuários.

  filteringPredicate
  O predicado de filtragem configura as condições de eventos que devem ser monitoradas para cada tipo de evento. Ele é formado por:

attributeFilter

O filtro do atributo é um atributo específico do tipo de evento que deve ser monitorado:

– É definido pelos seguintes elementos:

nome O atributo, ou nome da propriedade, do tipo de evento que deve ser monitorado. Consulte “Provedores e Definições de Eventos” na página 651 para obter uma lista de nomes de propriedades para cada tipo de evento.

operator

Pode ser:
- `eq` (igual a)
- `ne` (não igual a)
- `ge` (igual ou maior que)
- `le` (igual ou menor que)
- `range` (intervalo)

– Inclui um ou mais:

valor O valor no qual o operador deve ser correspondido.

Nota: Cada tipo de evento possui um número de atributos ou nomes de propriedade obrigatórios. Nem todos os nomes de propriedades obrigatórios possuem valores padrão. Todos os nomes de propriedades obrigatórios sem um valor padrão devem ter um predicado de filtragem definido.

correlationAttributes

Os atributos de correlação fornecem uma maneira de direcionar a regra para criar uma cópia da regra separada para cada grupo de eventos que compartilham características comuns. Em geral, cada regra ativa possui uma cópia da regra, que é executada no servidor de processamento de eventos. Entretanto, às vezes, a mesma regra é necessária para diferentes grupos de eventos, que geralmente estão relacionados a diferentes grupos de recursos. Utilizar um ou mais atributos de correlação é um método de direcionar uma regra para criar uma cópia da regra separada para cada grupo de eventos com características comuns. Utilize com os tipos de regra `set` e `sequence`.

Você pode especificar um ou mais atributos. Cada um é definido por:

attribute name=" "

O atributo do objeto que você está correlacionando.

action A ação que deve ser acionada se o evento for detectado. As definições de regras de eventos com eventos mas sem ações são aceitas sintaticamente, embora elas possam não ter significância prática. É possível salvar tais regras como rascunho e incluir ações posteriormente antes delas serem implementadas.

- Isto é definido pelos seguintes atributos necessários:

  actionProvider

  O nome do provedor de ação que pode implementar uma ou mais ações configuráveis. Os provedores de ação disponíveis no momento da instalação são:

  GenericAction

  Executa comandos não Tivoli Workload Scheduler. Os

comandos são executados no mesmo computador em que o processador de eventos é executado.

MailSender
Conecta a um servidor SMTP para enviar um e-mail.

MessageLogger
Registra a ocorrência de uma situação em um banco de dados de auditoria interno.

TECEventForwarder
Encaminha o evento para um servidor Tivoli Enterprise Console (TEC) externo ou para qualquer outro aplicativo capaz de atender aos eventos no formato TEC.

TWSAction
Executa um dos seguintes comandos de `conman`:
- submit job (`sbj`)
- submit job stream (`sbs`)
- submit command (`sbd`)
- reply to prompt (`reply`)

TWSForZosAction
Inclui uma ocorrência de aplicativo (fluxo de tarefas) no plano atual IBM Tivoli Workload Scheduler for z/OS. Esse provedor é para uso nas configurações de planejamento de ponta a ponta do Tivoli Workload Scheduler.

A descrição de aplicativo da ocorrência a ser incluída deve existir no banco de dados AD database do IBM Tivoli Workload Scheduler for z/OS.

actionType
Especifica o tipo de ação que deve ser acionada quando um evento especificado é detectado. Cada ação pode ser referida como um provedor de ação. A tabela a seguir lista os tipos de ação por provedor de ação. Para ver as propriedades de cada tipo de evento, acesse Apêndice A, "Evento de Automação e Definições de Ação da Carga de Trabalho Orientada a Evento", na página 651.

Tabela 53. Tipos de Ação por Provedor de Ação.

Provedor de Ação	Tipos de Ação
GenericAction	RunCommand
MailSender	SendMail
MessageLogger	PostOperatorMessage
TECEventForwarder	TECFWD
TWSAction	reply (ReplyPrompt)
	sbd (SubmitAdHocJob)
	sbj (SubmitJob)
	sbs (SubmitJobStream)
TWSForZosAction	AddJobStream

responseType
Especifica quando a ação deve ser executada. Os valores podem ser:

onDetection
A ação é iniciada assim que todos os eventos definidos na regra tiverem sido detectados. Aplica-se a todos os tipos de regra. Consulte também "Notas sobre a operação de regras" na página 132.

onTimeOut
A ação é iniciada após o tempo especificado em

`timeInterval` ter expirado mas nem todos os eventos definidos na regra terem sido recebidos. Aplica-se somente às regras `set` e `sequence`.

Observe que as ações de tempo limite não serão executadas se você não especificar um intervalo de tempo. O planejador, entretanto, permitirá que você salve as regras de eventos nas quais as ações de tempo limite foram definidas sem a especificação de um intervalo de tempo, pois será possível configurar o intervalo de tempo posteriormente. Até lá, somente as ações com o tipo de resposta `onDetection` serão processadas.

As ações de tempo limite para as quais um intervalo de tempo não foi definido são executadas apenas quando as regras são desativadas. Uma regra de evento é desativada em um dos dois casos:
– O comando `planman deploy -scratch` é emitido
– A regra é modificada (em seguida, é desativada assim que o comando `planman deploy` é executado)

Em qualquer um dos casos, a regra é primeiro desativada e depois reativada. Nesse momento, todas as ações pendentes são executadas.

- Inclui os seguintes atributos opcionais:
  descrição
  Uma descrição da ação. Pode ter até 120 caracteres.

  Lembre-se de gravar quaisquer caracteres XML especiais que você possa utilizar na notação de XML especial, tal como:
  – `&` para &
  – `>` para >
  – `<` para <
  – `"` para "

  e assim por diante.

  scope Um ou mais atributos de qualificação que descrevem a ação. Pode ter até 120 caracteres. O escopo é gerado automaticamente a partir do que é definido no XML. Ele não pode ser especificado pelos usuários.
- Inclui uma lista de um ou mais parâmetros ou nomes de propriedades. Todos os tipos de ação possuem pelo menos um parâmetro obrigatório. Cada parâmetro é definido por:
  parameter name=" "
  Consulte “Provedores e Definições de Ação” na página 661 para obter uma lista de parâmetros, ou nomes de propriedades, disponíveis para cada tipo de ação.

  valor Consulte “Provedores e Definições de Ação” na página 661 para obter uma lista dos possíveis valores ou tipos de valores.

  Você pode utilizar a substituição de variável. Isto significa que, quando você definir os parâmetros de ação, poderá utilizar os nomes de propriedades dos eventos que acionam a regra de evento para substituir o valor do nome da propriedade da ação. Para fazer isso, grave o valor do parâmetro da ação que pretende substituir de uma das duas seguintes maneiras:
  – `${event.property}`

Substitui o valor no estado em que se encontra. Isso é útil para transmitir as informações para uma ação que funciona programaticamente com elas, por exemplo, o `schedTime` de um fluxo de tarefas.

– `%{event.property}`

  Substitui o valor formatado em formato legível a humanos. Isso é útil para passar as informações para uma ação que encaminha essas informações a um usuário, por exemplo para formatar o `schedTime` de um fluxo de tarefas no corpo de um e-mail.

Em que:

evento
: É o nome configurado para o acionamento de `eventCondition`.

propriedade
: É o nome de `attributeFilter` no predicado de filtragem da condição do evento de acionamento. O valor obtido pelo filtro do atributo quando a regra é acionada é substituído como um valor de parâmetro na definição da ação antes que ela seja executada.

  Observe que você também pode utilizar a substituição de variável quando nenhum `attributeFilter` foi especificado para um atributo e quando o atributo for de leitura.

Por exemplo, a tarefa de uma regra de evento é detectar quando qualquer uma das tarefas que possuem um nome que começa com `job15` é finalizada com erro e, quando isso acontece, enviar essa tarefa novamente. A seção `eventCondition` da regra é codificada conforme a seguir:

```
<eventCondition
      name="event1"
      eventProvider="TWSObjectsMonitor"
      eventType="JobStatusChanged">
  <filteringPredicate>
     <attributeFilter
           name="JobName"
           operator="eq">
        <value>job15*</value>
     </attributeFilter>
     <attributeFilter
           name="Workstation"
           operator="eq">
        <value>*</value>
     </attributeFilter>
     <attributeFilter
           name="Status"
           operator="eq">
        <value>Error</value>
     </attributeFilter>
  </filteringPredicate>
</eventCondition>
```

Curingas (* para diversos caracteres ou ? para caracteres únicos) são usados para generalizar a condição de evento que você deseja para todas as instâncias da tarefa cujo nome começa com `job15` e para suas estações de trabalho associadas. A substituição da variável é utilizada na seção `action` para enviar novamente a tarefa específica que finalizou com erro na mesma estação de trabalho. A seção `action` é:

```
<action
      actionProvider="TWSAction"
      actionType="sbj"
      responseType="onDetection">
  <description>Submit again the job that ended in error</description>
      <parameter name="JobDefinitionName">
        <value>${event1.JobName}</value>
      </parameter>
      <parameter name="JobDefinitionWorkstationName">
        <value>${event1.Workstation}</value>
      </parameter>
</action>
```

Exemplos

JOB7 possui uma dependência de arquivo em DAILYOPS.XLS. Assim que o arquivo é recebido, JOB7 deve iniciar o processamento do arquivo. Os seguintes controles de regras que JOB7 inicia dentro de um minuto após a transferência de DAILYOPS.XLS são concluídos. Se isso não acontecer, um e-mail será enviado ao operador evening. Isto é realizado pela definição de duas condições de eventos seqüenciais que precisam monitorar:

1. O primeiro evento que aciona a regra é a criação do arquivo DAILYOPS.XLS na estação de trabalho à qual ele deve ser transferido. Assim que este evento for detectado, uma instância da regra será criada e uma contagem com intervalo de um minuto será iniciada para detectar a próxima condição de evento.
2. O segundo evento é o envio de JOB7. Se este evento falhar em ser detectado no intervalo de tempo especificado, a regra terá o tempo limite excedido e a ação SendMail será iniciada.

```
<?xml version="1.0"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules"
      xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules
      EventRules.xsd">
   <eventRule
         name="sample_rule"
         ruleType="sequence"
         isDraft="no">
      <description>An email is sent if job JOB7 does not start
within
                  a minute after file DAILYOPS.XLS is created</description>
      <timeZone>America/Indianapolis</timeZone>
      <validity
            from="2007-01-01"
            to="2007-12-31" />
      <activeTime start="20:00:00" end="22:00:00" />
      <timeInterval
            amount="60"
            unit="seconds" />
      <eventCondition
            name="DAILYOPS_FTPed_event"
            eventProvider="FileMonitor"
            eventType="FileCreated">
         <filteringPredicate>
            <attributeFilter
                  name="FileName"
                  operator="eq">
               <value>c:/dailybus/DAILYOPS.XLS</value>
            </attributeFilter>
            <attributeFilter
                  name="Workstation"
                  operator="eq">
               <value>ACCREC03</value>
            </attributeFilter>
            <attributeFilter
                  name="SampleInterval"
                  operator="eq">
               <value>300</value>
            </attributeFilter>
         </filteringPredicate>
      </eventCondition>
      <eventCondition
            name="job_JOB7_problem_event"
            eventProvider="TWSObjectsMonitor"
            eventType="JobSubmit">
        <filteringPredicate>
           <attributeFilter
                 name="JobStreamWorkstation"
                 operator="eq">
              <value>ACCREC03</value>
           </attributeFilter>
```

```
            <attributeFilter
                  name="Workstation"
                  operator="eq">
               <value>ACCREC03</value>
            </attributeFilter>
            <attributeFilter
                  name="JobStreamName"
                  operator="eq">
               <value>JSDAILY</value>
            </attributeFilter>
            <attributeFilter
                  name="JobName"
                  operator="eq">
               <value>JOB7</value>
            </attributeFilter>
         </filteringPredicate>
      </eventCondition>
      <action
            actionProvider="MailSender"
            actionType="SendMail"
            responseType="onTimeOut">
         <description>Send email to evening operator stating job did not start</description>
         <parameter name="To">
            <value>eveoper@bigcorp.com</value>
         </parameter>
         <parameter name="Subject">
            <value>Job JOB7 failed to start within scheduled time on
                   workstation ACCREC03.</value>
         </parameter>
      </action>
   </eventRule>
</eventRuleSet>
```

Observe que o escopo não mostra a primeira vez que a regra é definida.

Para obter mais exemplos de regra de eventos, consulte os “Exemplos de Regras de Eventos” na página 128.

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

o Guia do Usuário do Console do Dynamic Workload, seção Criando uma Regra de Evento.

Definindo os aplicativos de carga de trabalho

É possível usar oaplicativos de carga de trabalho para padronizar uma solução de automação de carga de trabalho para que a solução possa ser reutilizada em um ou mais ambientes do Tivoli Workload Scheduler automatizando, portanto, os processos de negócios.

Os Aplicativos de carga de trabalho são definidos em um ambiente do Tivoli Workload Scheduler, referido como o ambiente de origem usando a interface gráfica com o usuário do Workload Designer acessível a partir do Dynamic Workload Console. Você cria um novo modelo do aplicativo de carga de trabalho e, em seguida, inclui um ou mais fluxos de tarefas nele. Para reproduzir a solução de automação de carga de trabalho em outro ambiente do Tivoli Workload Scheduler, o modelo do aplicativo de carga de trabalho é exportado e, em seguida, após algumas customizações manuais, ele é importado para o ambiente de destino.

O processo de exportação produz um arquivo compactado que contém todos os arquivos e informações necessárias para a execução da carga de trabalho em um ambiente do Tivoli Workload Scheduler diferente. O arquivo compactado contém:

Um arquivo de definições
Um arquivo XML, *aplicativo de carga de trabalhotemplatename*_Definitions.UTF8.xml, que contém as definições de todos os objetos exportados. Estas definições serão implementadas no ambiente de destino para que seja possível preencher o banco de dados de destino com os mesmos objetos existentes no ambiente de origem.

Um arquivo de mapeamento
Um arquivo de mapeamento, *aplicativo de carga de trabalhotemplatename*_Mapping.UTF8.properties, necessário para tratar os objetos que são dependentes da topologia do ambiente e que não podem ser reproduzidos sem alguma customização manual. O usuário de destino modificará o arquivo substituindo os nomes dos objetos no ambiente de origem pelos nomes que esses objetos têm no ambiente de destino.

Um arquivo de informações de referência
Um arquivo de informações de referência, *aplicativo de carga de trabalhotemplatename*_SourceEnv_reference.txt, contendo as definições das estações de trabalho usadas no modelo do aplicativo de carga de trabalho e outras informações que podem ser úteis para mapear corretamente o ambiente de origem para o ambiente de destino para permitir a execução do aplicativo de carga de trabalho.

Os Aplicativos de carga de trabalho são gerenciados com a interface com o usuário da linha de comandos do **wappman**. Para importar um aplicativo de carga de trabalho para um ambiente de destino, o arquivo de mapeamento customizado e o arquivo de definições são transmitidos para o comando wappman como entrada para que todos os objetos sejam criados automaticamente no banco de dados do Tivoli Workload Scheduler do ambiente de destino. Consulte “Comando wappman” na página 335 para obter a sintaxe de comando completa e o uso.

Criando um Modelo do aplicativo de carga de trabalho

Como definir um modelo do aplicativo de carga de trabalho usando o Dynamic Workload Console.

Para assegurar que a solução de automação de carga de trabalho possa ser facilmente reproduzida em outro ambiente, existem algumas melhores práticas a serem consideradas ao criar o modelo do aplicativo de carga de trabalho:

Definições de tarefas
As tarefas que se referem a elementos que são dependentes do ambiente ou da topologia em que eles residem, como tarefas de serviço da web, tarefas de transferência de arquivos e tarefas do banco de dados, para citar algumas, devem usar variáveis ao especificar estes elementos, como credenciais, caminhos e números de portas. As variáveis podem ser gerenciadas no arquivo de mapeamento para que os valores corretos possam ser designados à variável.

Nomes de estações de trabalho
Quando tarefas e fluxos de tarefas são extraídos do aplicativo de carga de trabalho durante o processo de exportação, os nomes das estações de trabalho são extraídos conforme são localizados no ambiente de origem. Os nomes significativos ou uma convenção de nomenclatura padronizada podem simplificar o processo de mapeamento.

Usuários
Os usuários também são extraídos conforme são localizados no ambiente de origem. Se o mesmo usuário não estiver presente no ambiente de origem e de destino, variáveis deverão ser usadas para especificar o usuário.

Arquivo de Mapeamento
O arquivo de mapeamento deve ser mantido após a execução do processo de importação. Ele pode ser útil no caso em que você deseja substituir um aplicativo de carga de trabalho ou atualizá-lo, fazendo as mudanças necessárias no arquivo de mapeamento.

Tabela de variável do fluxo de tarefas
Todas as variáveis usadas para representar genericamente objetos no aplicativo de carga de trabalho devem ser incluídas em uma tabela de variáveis específica relacionada ao fluxo de tarefas no aplicativo de carga de trabalho. Isto permite a customização do fluxo de tarefas para refletir o ambiente de destino por meio do arquivo de mapeamento. Evite associar a tabela de variáveis padrão a um fluxo de tarefas. A tabela de variáveis padrão é extraída como qualquer outra tabela e precisará ser renomeada, caso contrário, o processo de importação falhará, porque já existe uma tabela com o mesmo nome. O ambiente de destino já possui uma tabela de variáveis padrão, MAIN_TABLE, definida.

Tabela de variável de ciclo de execução
Todas as variáveis usadas para representar genericamente objetos no aplicativo de carga de trabalho devem ser incluídas em uma tabela de variáveis específica relacionada ao ciclo de execução no aplicativo de carga de trabalho. Isto permite a customização do ciclo de execução para refletir o ambiente de destino por meio do arquivo de mapeamento. Evite associar a tabela de variáveis padrão a um ciclo de execução. A tabela de variáveis padrão é extraída como qualquer outra tabela e precisará ser renomeada, caso contrário, o processo de importação falhará, porque já existe uma tabela com o mesmo nome. O ambiente de destino já possui uma tabela de variáveis padrão, MAIN_TABLE, definida.

No Workload Designer, é possível criar o modelo de uma carga de trabalho que pode então ser importada e executada em outro ambiente. É possível criar um modelo do aplicativo de carga de trabalho contendo um ou mais fluxos de tarefas com todas as tarefas relacionadas e dependências internas ou externas (como arquivos, recursos, prompts) para ter um fluxo de trabalho autocontido. É possível então exportar o modelo do aplicativo de carga de trabalho para implementar e executá-lo em outro ambiente. Para criar um modelo do aplicativo de carga de trabalho, execute o seguinte procedimento:

1. Na barra de ferramentas de navegação, clique em **Administração** > **Design de Carga de Trabalho** > **Gerenciar Definições de Carga de Trabalho**
2. Especifique o nome de um mecanismo distribuído. O Workload Designer é aberto.
3. Na área de janela Lista de Trabalho, selecione **Novo** > **Modelo de Aplicativo de Carga de Trabalho**. O modelo do aplicativo de carga de trabalho é criado na visualização Detalhes e sua página de propriedades é exibida.
4. Na área de janela de propriedades, especifique os atributos para o modelo do aplicativo de carga de trabalho que está sendo criado:

Per. Campo obrigatório que contém o nome do modelo do aplicativo de carga de trabalho. O comprimento máximo é de 80 caracteres.

Descrição
Texto descritivo opcional para ajudar usuários do aplicativo de carga de trabalho a entenderem o propósito e características do aplicativo de carga de trabalho. O comprimento máximo é de 120 caracteres.

Provedor
Campo opcional que especifica o criador do modelo do aplicativo de carga de trabalho. Ele pode ser útil para permitir que usuários do aplicativo de carga de trabalho saibam quem o criou e o forneceu. O comprimento máximo é de 120 caracteres.

5. Na visualização Detalhes, clique com o botão direito no modelo do aplicativo de carga de trabalho e clique em **Incluir Fluxo de Tarefas** para incluir fluxos de tarefas nele.
6. No diálogo de consulta, selecione os fluxos de tarefas que você deseja incluir. Juntamente com os fluxos de tarefas, as dependências correspondentes também são automaticamente incluídas no modelo do aplicativo de carga de trabalho.
7. Clique em **Salvar** para salvar o modelo do aplicativo de carga de trabalho no banco de dados.
8. Clique com o botão direito do mouse no modelo do aplicativo de carga de trabalho e clique em **Exportar** para produzir um arquivo compactado, denominado ***aplicativo de carga de trabalho nome do modelo*.zip**, contendo todos os arquivos e informações necessários para permitir que a carga de trabalho também seja executada em outro ambiente.

O arquivo compactado contém:

*Nome do modelo do aplicativo de carga de trabalho*_**Definitions.UTF8.xml**
Arquivo XML que contém as definições de todos os objetos exportados. Essas definições serão implementadas no ambiente de destino para preencher o banco de dados de destino com os mesmos objetos existentes no ambiente de origem. Os objetos no arquivo de definição podem permanecer como estão ou é possível optar por renomeá-los. Se um objeto não tiver uma definição no arquivo de definição, por exemplo, uma estação de trabalho, então, no tempo de importação, um objeto correspondente não será criado no ambiente de destino. A expectativa é que esse objeto já esteja presente no ambiente de destino, portanto, para esses tipos de objeto, você deverá mapeá-los no arquivo de mapeamento.

*Nome do modelo do aplicativo de carga de trabalho*_**Mapping.UTF8.properties**
O arquivo de mapeamento usado para substituir os nomes dos objetos no ambiente de origem pelos nomes que esses objetos terão no ambiente de destino. Os objetos que serão criados no ambiente de destino podem ser criados com os mesmos nomes daqueles no ambiente de origem ou é possível especificar um nome diferente nesse arquivo.

*Nome do modelo do aplicativo de carga de trabalho*_**SourceEnv_reference.txt**
Informações de referência contendo as definições das estações de trabalho usadas no modelo do aplicativo de carga de trabalho e outras informações que podem ser úteis para mapear corretamente o ambiente de origem para o ambiente de destino para permitir que o aplicativo de carga de trabalho seja executado.

O pacote compactado deve então ser importado para o ambiente de destino no qual o aplicativo de carga de trabalho será implementado, criando, assim, todos os

objetos necessários no ambiente de destino. No ambiente de destino, o arquivo *name*_Mapping.UTF8.properties do aplicativo de carga de trabalho deve ser editado manualmente, usando um editor de texto genérico, especificando os nomes dos objetos conforme eles são definidos no ambiente de destino (por exemplo, os nomes das estações de trabalho nas quais os fluxos de tarefas são executados). A operação de importação deve ser executada no ambiente de destino usando a linha de comandos. Para obter detalhes adicionais, consulte as seções do *Guia do Usuário e Referência do Tivoli Workload Scheduler* sobre aplicativos de carga de trabalho e o comando wappman.

Capítulo 9. Gerenciando Objetos no Banco de Dados - Composer

Este capítulo descreve como utilizar o programa de linha de comandos **composer** para gerenciar os objetos de planejamento no banco de dados do Tivoli Workload Scheduler. Ela está dividida nas seguintes seções:

- "Configurando o Programa fe Linha de Comandos composer"
- "Executando Comandos a partir do Composer" na página 281
- "Composer commands" na página 284

Configurando o Programa fe Linha de Comandos composer

O programa de linha de comandos **composer** gerencia objetos de planejamento no banco de dados.

Você deve instalar o recurso do Cliente da Linha de Comando do Tivoli Workload Scheduler nos agentes tolerantes a falhas e sistemas fora da rede do Tivoli Workload Scheduler para utilizar o programa da linha de comandos **composer**.

Configurando o Ambiente do Composer

Esta seção descreve como configurar o ambiente do composer.

Saída do Terminal

As variáveis de shell denominadas ***MAESTROLINES*** e ***MAESTROCOLUMNS*** determinam a saída para seu computador. Se uma delas não estiver definida, as variáveis shell padrão, ***LINES*** e ***COLUMNS***, serão utilizadas. No final de cada pagina de tela, o composer solicita para continuar. Se ***MAESTROLINES*** (ou ***LINES***) estiver definido como zero ou um número negativo, o composer não será pausado no final de uma página.

Dependendo do valor configurado na variável local ***MAESTROCOLUMNS***, dois conjuntos diferentes de informações são exibidos sobre o objeto selecionado. Há duas possibilidades:

- MAESTROCOLUMNS < 120 caracteres
- MAESTROCOLUMNS > 120 caracteres

O valor configurado na variável local ***MAESTROCOLUMNS*** não pode ser mais alto que 1024.

Consulte a Tabela 60 na página 297 e a Tabela 61 na página 308 para saber os diferentes formatos de saída.

Saída Offline

A opção **;off-line** nos comandos do composer é utilizada para imprimir a saída de um comando. Ao incluí-la, as seguintes variáveis controlam a saída:

Variáveis do Windows:

MAESTROLP
: Especifica o arquivo no qual a saída de um comando é gravada. O padrão é **stdout**.

© Copyright IBM Corp. 1999, 2014

MAESTROLPLINES
Especifica o número de linhas por página. O padrão é 60.

MAESTROLPCOLUMNS
Especifica o número de caracteres por linha. O padrão é 132.

Variáveis do UNIX:
A opção **;off-line** nos comandos do composer é utilizada para imprimir a saída de um comando. Ao incluí-la, as seguintes variáveis shell controlam a saída:

MAESTROLP
Especifica o destino da saída de um comando. Defina-o como um dos seguintes:

> *file* Redireciona a saída para um arquivo, sobrescrevendo seu conteúdo. Se o arquivo não existir, ele será criado.

>> *file* Redireciona a saída para um arquivo, anexando-a ao final do arquivo. Se o arquivo não existir, ele será criado.

| *command*
Canaliza a saída para um comando do sistema ou processo. O comando do sistema é executado independentemente da geração da saída.

|| *command*
Canaliza a saída para um comando do sistema ou processo. O comando do sistema não será executado, se não houver saída.

O valor padrão para ***MAESTROLP*** é **| lp -tCONLIST**, que direciona a saída do comando para a impressora e coloca o título "CONLIST" na página de banner da impressão.

MAESTROLPLINES
Especifica o número de linhas por página. O padrão é **60**.

MAESTROLPCOLUMNS
Especifica o número de caracteres por linha. O padrão é **132**.

As variáveis devem ser exportadas antes da execução do **composer**.

O Editor do Composer

Vários comandos do composer abrem automaticamente um editor de texto. É possível selecionar qual editor deseja que o composer utilize.

Além disso, no Windows e no UNIX, é possível configurar a variável de ambiente `XMLEDIT` para apontar para um editor XML de sua escolha para editar definições de regra de evento. O editor XML é aberto automaticamente toda vez que você executa os comandos `composer add`, `new` ou `modify` em uma regra de evento.

Windows:
No Windows, o **Notepad** é utilizado como o editor padrão. Para alterar o editor, configure a variável de ambiente *EDITOR* com o caminho e nome do novo editor antes de executar o **composer**.

UNIX:
Vários comandos que podem ser emitidos a partir do **composer** abrem automaticamente um editor de texto. O tipo de editor é determinado pelo valor de duas variáveis shell. Se a variável ***VISUAL*** estiver definida, ela definirá o editor, caso contrário, a variável ***EDITOR*** definirá o editor. Se nenhuma das duas estiver definida, um editor **vi** será aberto.

Selecionando o Prompt do composer no UNIX

O prompt de comandos do **composer** é definido no arquivo *TWS_home*/localopts. O prompt de comandos padrão é um traço (-). Para selecionar um prompt diferente, edite a opção ***composer prompt*** no arquivo localopts e altere o traço. O prompt pode ter até dez caracteres, não incluindo o sinal de sustenido (#) final necessário.

```
#--------------------------------------------------------------------------
# Atributos de formato customizado
#
date format =    1   # The possible values are 0-ymd, 1-mdy,
2-dmy, 3-NLS.
composer prompt =   -
conman prompt =    %
switch sym prompt =  <n>%
#--------------------------------------------------------------------------
```

Para obter informações adicionais sobre o arquivo de configuração localopts, consulte o *Guia de Administração do IBM Tivoli Workload Scheduler*.

Executando o Programa composer

Para configurar seu ambiente para utilizar o**composer**, configure as variáveis *PATH* e *TWS_TISDIR* executando um dos seguintes scripts:

No UNIX:

- **. ./***TWS_home*/**tws_env.sh** para shells Bourne e Korn
- **. ./***TWS_home*/**tws_env.csh** para shells C

No Windows:

- *TWS_home***tws_env.cmd**

Em seguida, utilize a sintaxe a seguir para executar comandos a partir da interface com o usuário **composer**:

composer [**-file** *filename*][***connection_parameters***] ["*command*[**&**[*command*]][...]"]

em que:

-file *filename*
: Indica um arquivo de propriedades customizado alternativo que contém as configurações para os parâmetros de conexão, utilizado no lugar dos arquivos **useropts** e **localopts**.

connection_parameters
: Se estiver usando**composer** a partir do gerenciador de domínio principal, os parâmetros de conexão foram configurados na instalação e não precisam ser fornecidos, a menos que você não deseje usar os valores padrão.

  Se estiver usando**composer** a partir do cliente da linha de comandos em outra estação de trabalho, os parâmetros de conexão podem ser fornecidos por um ou mais destes métodos:

  - Armazenados no arquivo localopts
  - Armazenados no arquivo useropts
  - Fornecidos ao comando em um arquivo de parâmetro
  - Fornecidos ao comando como parte da sequência de caracteres de comando

  Para obter uma visão geral dessas opções, consulte “Configurando Opções para Utilização das Interfaces com o Usuário” na página 53. Para obter detalhes integrais dos parâmetros de configuração, consulte o tópico sobre como configurar o acesso do cliente de linha de comandos no *Tivoli Workload Scheduler: Administration Guide*.

O programa de linha de comandos **composer** é instalado automaticamente ao instalar ogerenciador de domínio principal. Ele deve ser instalado separadamente sobre uma estação de trabalho de agente do Tivoli Workload Scheduler ou de forma independente em um nó fora da rede do Tivoli Workload Scheduler. O recurso que instala o programa de linha de comandos **composer** é denominado*Cliente da Linha de Comandos*. Para obter informações sobre como instalar o recurso*Cliente da Linha de Comandos*, consulte o *IBM Tivoli Workload Scheduler: Planning and Installation Guide*.

Depois de instalado, é possível utilizar a linha de comandos **composer** no modo de *lote* e *interativo*.

Ao executar o**composer** no modo*interativo*, você primeiro ativa o programa de linha de comandos **composer** e, em seguida, a partir do prompt da linha de comandos **composer**, você executa comandos um por vez, por exemplo:

```
composer –username admin2 –password admin2pwd
   add myjobs.txt
   create myjobs.txt from jobs=@
```

Ao executar **composer** no modo*em lote*, você ativa o programa de linha de comandos **composer** especificando como parâmetro de entrada o comando a ser emitido. Quando o comando é processado, o programa de linha de comandos **composer** sai, por exemplo:

```
composer –f "c:\TWS\network\mylocalopts" add myjobs.txt
```

Nota: Se utilizar o modo de batch para emitir mais de um comando a partir do **composer**, certifique-se de gerenciar o caractere ; de uma das seguintes formas:

- Utilizando aspas duplas, por exemplo:

  ```
  composer "delete dom=old_domain; noask"
  ```

- Utilizando um caractere de espaço, por exemplo:

  ```
  composer delete dom=old_domain noask
  ```

- Fazendo escape do caractere ;, por exemplo:

  ```
  composer delete dom=old_domain \; noask
  ```

Outros exemplos de como utilizar o comando, supondo que os parâmetros de conexão estejam configurados nos scripts de configuração local, são os seguintes:

- Executa os comandos **print** e **version** e sai:

  ```
  composer "p parms&v"
  ```

- Executa os comandos **print** e **version** e, em seguida, solicita um comando:

  ```
  composer "p parms&v&"
  ```

- Lê comandos do `cmdfile`:

  ```
  composer < cmdfile
  ```

- Canaliza comandos do `cmdfile` para o **composer**:

  ```
  cat cmdfile | composer
  ```

Nota: Nas estações de trabalho Windows, se o UAC (User Account Control) estiver ligado e a lista de exceções do UAC não contiver o arquivo `cmd.exe`, você deve abrir a shell do prompt de comandos do DOS com a opção `"Executar como Administrador"` para executar o**composer** em sua estação de trabalho como um usuário genérico diferente do Administrador ou o usuário do Tivoli Workload Scheduler.

Caracteres de Controle

É possível digitar os seguintes caracteres de controle no modo conversacional para interromper o **composer** se as definições de ***stty*** estiverem configuradas para isso.

Ctrl+c O **composer** pára a execução do comando atual na próxima etapa que puder ser interrompida e retorna um prompt de comandos.

Ctrl+d O **composer** sai após a execução do comando atual.

Executando Comandos a partir do Composer

Os comandos do **Composer** consistem nos seguintes elementos:

commandname selection arguments

em que:

commandname
Especifica o nome do comando.

selection
Especifica o objeto ou conjunto de objetos sobre os quais a ação será tomada.

argumentos
Especifica os argumentos do comando.

Filtros e Curingas

No Tivoli Workload Scheduler **composer** é possível utilizar curingas e filtros ao emitir alguns comandos específicos para filtrar os objetos de planejamento definidos no banco de dados. Os curingas que podem ser utilizados a partir do **composer** são:

@ Substitui um ou mais caracteres alfanuméricos.

? Substitui um caractere alfanumérico.

Para procurar ocorrências com nomes que contêm @ ou ?, certifique-se de utilizar o caractere barra invertida \ antes de @ ou ? para escapá-los para que não sejam interpretados como curingas. De forma semelhante, o caractere barra invertida deve ter como prefixo outro caractere barra invertida para ser interpretado como uma ocorrência a ser localizada. Os exemplos a seguir esclarecem essas regras, que também se aplicam ao especificar as cadeias de procura utilizando a palavra-chave **filter**.

S@E Procurar todas as cadeias iniciadas por S e terminadas por E, independentemente de seu comprimento.

S?E Procurar todas as cadeias iniciadas por S e terminadas por E, cujo comprimento é três caracteres.

S\@E Procurar uma correspondência exata com a cadeia S@E.

S\?E Procurar uma correspondência exata com a cadeia S?E.

S\\E Procurar uma correspondência exata com a cadeia S\E.

Os comandos que podem ser emitidos a partir do composer e que suportam filtragem são:

- display
- create
- excluir
- lista
- lock

- modify
- print
- destravar

A sintaxe utilizada para filtrar objetos ao emitir um desses comandos é a seguinte:

command_name type_of_object=selection; [*option;*] [**filter** *filter_keyword*=selection [...]]

A Tabela 54 mostra os objetos de planejamento que é possível filtrar ao emitir os comandos listados acima e para cada objeto, cujos campos podem ser filtrados (em *itálico*) ou cuja chave (em **negrito)** é utilizada para filtrar seus campos:

Tabela 54. Critérios de Filtragem de Objetos de Planejamento

Objeto de planejamento	Palavras-chave e campos de filtro que podem ser filtrados	Descrição	Exemplo
estação de trabalho	*workstationname*	Aplica o comando às estações de trabalho cujo nome satisfaz os critérios.	list ws=p@
	domínio	Aplica o comando às estações de trabalho que pertencem a um domínio.	mod ws=@; filter **domain**=dom1
	vartable	Aplica o comando às estações de trabalho que se referem à tabela de variáveis especificada.	mod ws=@; filter **vartable**=table2
domínio	*domainname*	Aplica o comando aos domínios cujo nome satisfaz os critérios.	display dom=dom?
	parent	Aplica o comando aos domínios cujo domínio pai satisfaz os critérios.	list dom=@; filter **parent**=rome
prompt	*promptname*	Aplica o comando aos prompts globais cujo nome satisfaz os critérios.	lock prompt=p@
usuário	*workstationname# username*	Aplica o comando aos usuários cujo identificador satisfaz os critérios.	list users=cpu1#operator?
recurso	*workstationname# resourcename*	Aplica o comando aos recursos cujo identificador satisfaz os critérios.	print res=cpu?#operator?
variável	*variablename*	Aplica o comando aos parâmetros cujo nome satisfaz os critérios.	delete vb=mytable.myparm@
definição de tarefa	*jobname*	Aplica o comando às definições de tarefas cujo nome satisfaz os critérios.	mod jd=mycpu#myjob@
	RecoveryJob	Aplica o comando às tarefas cuja definição contém a definição da tarefa de recuperação especificada.	list jobdefinition=@; filter **RecoveryJob**=CPUA#job01
fluxo de tarefas	*workstationname# jobstreamname*	Aplica o comando às definições de tarefas cujo nome satisfaz os critérios.	mod js=mycpu#myjs@
	Calendar ou **FreeCalendar**	Aplica o comando aos fluxos de tarefa que contêm o calendário ou calendário de dias livres especificado no filtro.	list js=@#@; filter **Calendar**=cal1
	Jobdefinition	Aplica o comando aos fluxos de tarefas que contêm a definição de tarefa especificada no filtro.	list js=@#@; filter **jobdefinition**=CPUA#job01
	Resource	Aplica o comando aos fluxos de tarefas que se referem ao recurso especificado no filtro.	list js=@#@; filter **Resource**=cpu1#disk1
	Prompt	Aplica o comando aos fluxos de tarefas que se referem ao prompt especificado no filtro.	list js=@#@; filter **Prompt**=myprompt
	Vartable	Aplica o comando aos fluxos de tarefas que se referem à tabela de variáveis especificada no filtro. A tabela de variáveis pode ser especificada no ciclo de execução ou na seção de fluxo de tarefas.	list js=@#@; filter **Vartable**=table1
	Rcvartable	Aplica o comando aos ciclos de execução nos fluxos de tarefas que se referem à tabela de variáveis especificada no filtro.	list js=@#@; filter **Rcvartable**=table1
	Jsvartable	Aplica o comando aos fluxos de tarefas que se referem à tabela de variáveis especificada no filtro, independentemente do que seja especificado no ciclo de execução.	list js=@#@; filter **Jsvartable**=table1
regra de evento	*eventrulename*	Aplica o comando às regras de eventos que incluem uma ação em uma tarefa ou fluxo de tarefas específicos.	list er=@; filter js=accrecjs5

Tabela 54. Critérios de Filtragem de Objetos de Planejamento (continuação)

Objeto de planejamento	Palavras-chave e campos de filtro que podem ser filtrados	Descrição	Exemplo
vartable	*vartablename*	Aplica o comando às tabelas de variáveis cujo nome satisfaz os critérios.	list vartable=A@
	isdefault	Aplica o comando à tabela de variáveis padrão.	list vartable=A@; filter isdefault

É possível combinar mais de um filtro para o mesmo tipo de objeto, conforme mostrado no exemplo a seguir:

```
list js=@#@; filter Calendar=cal1 FreeCalendar=VACATIONS jobdefinition=CPUA#job01
```

A saída do comando é uma lista de fluxos de tarefas que utilizam o calendário `cal1`, calendário de dias livres `VACATIONS` e contém uma tarefa com a definição de tarefa `CPUA#job01`.

Delimitadores e caracteres especiais

A Tabela 55 lista caracteres que têm significados especiais nos comandos composer.

Tabela 55. Delimitadores e Caracteres Especiais para o Composer

Caractere	Descrição
&	Delimitador de comando. Consulte "Executando o Programa composer" na página 279.
;	Delimitador de argumento. Por exemplo: `;info;off-line`
=	Delimitador de valor. Por exemplo: `sched=sked5`
:!	Prefixos de comandos que transmitem o comando para o sistema. Esses prefixos são opcionais; se o composer não reconhecer o comando, ele será transmitido automaticamente para o sistema. Por exemplo: `!ls ou :ls`
<< >>	Colchetes de comentários. Os comentários podem ser colocados em uma única linha em qualquer lugar do fluxo de tarefas. Por exemplo: `schedule foo <<comment>> on everyday`

*	Prefixo do comentário. Quando este prefixo for o primeiro caractere de uma linha, toda a linha será um comentário. Quando o prefixo vier depois de um comando, o restante da linha será um comentário. Por exemplo:
	`*comment`
	ou
	`print& *comment`

>	Redireciona a saída do comando para um arquivo, sobrescrevendo seu conteúdo. Se o arquivo não existir, ele será criado. Por exemplo: `display parms > parmlist`
>>	Redireciona a saída de comando para um arquivo e anexa a saída no final do arquivo. Se o arquivo não existir, ele será criado. Por exemplo: `display parms >> parmlist`
|	Canaliza a saída de comando para um comando do sistema ou processo. O comando do sistema é executado independentemente da geração da saída. Por exemplo: `display parms | grep alparm`
||	Canaliza a saída de comando para um comando do sistema ou processo. O comando do sistema não será executado, se não houver saída. Por exemplo: `display parms || grep alparm`

Códigos de Retorno do Composer

Códigos de Retorno do Composer

Gerenciamento de códigos de retorno do Composer

Quando você executa um comando do composer, a linha de comandos pode mostrar um código de retorno de resultado. Para localizar o código de retorno, execute a ação a seguir:

Em sistemas operacionais Windows:

```
echo %ERRORLEVEL%
```

Em sistemas operacionais UNIX:

```
echo $?
```

A linha de comandos do Composer possui os códigos de retorno a seguir:

0 Comando concluído com êxito.

4 Comando concluído com um aviso.

8 Comando concluído com um erro.

16 O comando falhou.

32 O comando possui um erro de sintaxe.

Composer commands

A Tabela 56 lista os comandos **composer**.

Nota: Os nomes e as palavras-chave dos comandos podem ser digitados em caracteres maiúsculos ou minúsculos e podem ser abreviados com o mínimo de caracteres iniciais, o necessário para distingui-los exclusivamente uns dos outros. Alguns dos nomes de comandos também possuem formas abreviadas.

No entanto, há algumas abreviações, como **v**, que apontam para um comando específico, **version** neste caso, apesar de não identificarem exclusivamente esse comando na lista. Isso ocorre quando a abreviação é código permanente no produto e, portanto, incompatibilidades em chamar o comando errado são automaticamente ignoradas.

Tabela 56. Lista de Comandos do Composer

Comando	Nome Abreviado	Descrição	Consulte a página
incluir	**a**	Inclui uma definição de objetos de planejamento para o banco de dados a partir de um arquivo de texto.	"add" na página 287
authenticate	**au**	Altera as credenciais do usuário que está executando o **composer**.	"authenticate" na página 289
continuar	**c**	Ignora o próximo erro.	"continue" na página 290
create	**cr**	Extrai uma definição de objeto do banco de dados e grava a mesma em um arquivo de texto. Sinônimo para o comando **extract**.	"extract" na página 300
delete	**de**	Exclui objetos de planejamento.	"delete" na página 290
display	**di**	Exibe os detalhes do objeto de planejamento especificado.	"display" na página 294
editar	**ed**	Edita um arquivo.	"edit" na página 299
exit	**e**	Sai do **composer**.	"exit" na página 299
extract	**ext**	Extrai uma definição de objeto do banco de dados e grava a mesma em um arquivo de texto.	"extract" na página 300
help	**h**	Chama a ajuda online para um comando.	"help" na página 304
lista	**l**	Lista objetos de planejamento.	"list" na página 305
lock	**lo**	Bloqueia o acesso aos objetos de banco de dados.	"lock" na página 310
modificar	**m**	Modifica objetos de planejamento.	"modify" na página 314
novo		Cria um objeto de planejamento usando um arquivo de texto em que a definição do objeto é inserida online. Se você especificar o tipo de objeto de planejamento que deseja definir após o comando *new*, uma definição de objeto predefinida será gravada no arquivo de texto.	"new" na página 319
print	**p**	Imprime objetos de planejamento.	"display" na página 294
redo	**red**	Edita e executa o comando anterior novamente.	"redo" na página 321
rename	**rn**	Altera o nome do objeto.	"rename" na página 322
replace	**rep**	Substitui objetos de planejamento.	"replace" na página 324
system command		Chama um comando de sistema operacional.	"system command" na página 325
destravar	**b**	Libera o bloqueio do objeto de planejamento definido no banco de dados.	"unlock" na página 326
validate	**val**	Valida a integridade da sintaxe, da semântica e dos dados de uma definição de objeto.	"validate" na página 329
version	**v**	Exibe o banner do programa de linha de comandos **composer**.	"version" na página 330

Tabela 56. Lista de Comandos do Composer (continuação)

Comando	Nome Abreviado	Descrição	Consulte a página
`wkldapp`		Cria, substitui, exclui, exibe e lista um aplicativo de carga de trabalho.	"Comando wappman" na página 335

Verificação de Integridade Referencial

O Tivoli Workload Scheduler executa automaticamente as verificações referenciais para evitar falta de integridade nas definições de objetos no banco de dados toda vez que você executar comandos que criam, modificam ou excluem a definição de um objeto referenciado. Estas são as verificações executadas pelo produto:

- Toda vez que você utiliza um comando que cria um novo objeto no banco de dados, o Tivoli Workload Scheduler verifica se:
  - Um objeto do mesmo tipo e com o mesmo identificador já existe.
  - Os objetos referidos por esse objeto já existem no banco de dados.
- Toda vez que você executar um comando que modifica uma definição de objeto no banco de dados, o Tivoli Workload Scheduler verifica se:
  - A definição do objeto a ser modificada existe no banco de dados.
  - Os objetos referidos por esse objeto existem no banco de dados.
  - Para evitar inconsistências de integridade, a definição de objeto não aparece na definição de um objeto que pertence à cadeia de seus ascendentes.
- Toda vez que você executar um comando que exclui uma definição de objeto no banco de dados, o Tivoli Workload Scheduler verifica se:
  - A definição de objeto a ser excluída existe no banco de dados.
  - A definição de objeto a ser excluída não é referida por outros objetos definidos no banco de dados.

Observe que não há qualquer verificação de integridade referencial para regras de evento.

A Tabela 57 mostra, para cada tipo de objeto, os identificadores que são utilizados para identificar exclusivamente o objeto no banco de dados ao criar ou modificar definições de objetos:

Tabela 57. Identificadores de Objetos para Cada Tipo de Objeto Definido no Banco de Dados

Tipo de Objeto	Identificadores de Objetos
domínio	*domainname*
estação de trabalho	*workstationname* (verificado em estações de trabalho e classes de estação de trabalho)
classe de estação de trabalho	*workstationclassname* (verificado em estações de trabalho e classes de estação de trabalho)
calendário	*calendarname*
definição de tarefa	*workstationname* e *jobname*
usuário	*workstationname* e *username*
fluxo de tarefas	*workstationname* e *jobstreamname* e, se definido, *validfrom*
tarefa em um fluxo de tarefa	*workstationname* e *jobstreamname*, *jobname* e, se definido, *validfrom*
recurso	*workstationname* e *resourcename*
prompt	*promptname*
tabela de variáveis	*variabletablename*
variável	*variabletablename.variablename*
regra de evento	*eventrulename*

Em geral, a integridade referencial evita a exclusão de objetos quando eles são referenciados por outros objetos do banco de dados. No entanto, em alguns casos,

em que a exclusão de um objeto (por exemplo, uma estação de trabalho) sugere somente a atualização de um objeto referido (por exemplo, uma classe de estação de trabalho que o inclui), a exclusão pode ser permitida. A Tabela 58 mostra todos os casos em que um objeto referido pode ser excluído mesmo se outros objetos fizerem referência a ele:

Tabela 58. Atualização da Definição de Objeto na Exclusão do Objeto Referenciado

Objeto	Referências	Na Exclusão do Objeto Referenciado...
Dependência de Interligação de Redes	Estação de Trabalho	... remova a dependência da tarefa ou do fluxo de tarefas
Dependência Follows Externa	Fluxo de Tarefas	... remova a dependência da tarefa ou do fluxo de tarefas
	Tarefa	... remova a dependência da tarefa ou do fluxo de tarefas
Dependência Interna	Tarefa	... remova a dependência da tarefa ou do fluxo de tarefas
Classe de Estação de Trabalho	Estação de Trabalho	... remova a estação de trabalho da classe de estação de trabalho

A Tabela 59 descreve como o produto se comporta quando é solicitado que exclua um objeto referido por outro objeto com a utilização de um relacionamento específico:

Tabela 59. Referential integrity check when deleting an object from the database

Objeto a Ser Excluído	Referenciado pelo Objeto	Relacionamento	Regra de Exclusão
domínio A	domínio B	domínio A é pai do domínio B	Um erro especificando o relacionamento existente é exibido.
	estação de trabalho B	estação de trabalho b pertence à estação de trabalho A	Um erro especificando o relacionamento existente é exibido.
estação de trabalho A	estação de trabalho B	estação de trabalho A é host da estação de trabalho B	Um erro especificando o relacionamento existente é exibido.
	tarefa B	tarefa B está definida na estação de trabalho A	Um erro especificando o relacionamento existente é exibido.
	fluxo de tarefa B	fluxo de tarefa B está definido na estação de trabalho A	Um erro especificando o relacionamento existente é exibido.
	usuário B	usuário B está definido na estação de trabalho A	Um erro especificando o relacionamento existente é exibido.
	fluxo de tarefa B	estação de trabalho A funciona como agente de rede para dependências da interligação de redes configuradas no fluxo de tarefa B	A estação de trabalho A e a dependência de interligação de redes são excluídas
	fluxo de tarefa B	fluxo de tarefa B tem uma dependência de arquivo de um arquivo definido na estação de trabalho A	A estação de trabalho A e a dependência de arquivo são excluídas
	tarefa B no fluxo de tarefa B	estação de trabalho A funciona como agente de rede para dependências da interligação de redes configuradas na tarefa B	A estação de trabalho A e a dependência de interligação de redes são excluídas
	tarefa B no fluxo de tarefa B	tarefa B tem uma dependência de arquivo de um arquivo definido na estação de trabalho A	A estação de trabalho A e a dependência de arquivo são excluídas
	recurso B	recurso B está definido na estação de trabalho A	Um erro especificando o relacionamento existente é exibido.
	arquivo B	arquivo B está definido na estação de trabalho A	Um erro especificando o relacionamento existente é exibido.
	classe de estação de trabalho B	estação de trabalho A pertence à classe de estação de trabalho B	A estação de trabalho A e sua entrada na classe de estação de trabalho B são excluídas.
	tarefa B no fluxo de tarefa B	tarefa B contida no fluxo de tarefa B está definida na estação de trabalho A	Um erro especificando o relacionamento existente é exibido.
tarefa A	tarefa B	tarefa A é a tarefa de recuperação para a tarefa B	Um erro especificando o relacionamento existente é exibido.
	fluxo de tarefa B	tarefa A está contida no fluxo de tarefa B	Um erro especificando o relacionamento existente é exibido.
	fluxo de tarefa B	fluxo de tarefa B segue a tarefa A	A tarefa A e a dependência follows no fluxo de tarefa B são excluídas.
	tarefa B no fluxo de tarefa B	tarefa B segue a tarefa A	A tarefa A e a dependência follows na tarefa B são excluídas.
	regra de evento B	tarefa A está na definição de ação da regra de evento B (e não utiliza substituição de variáveis)	Um erro especificando o relacionamento existente é exibido.
calendário A	fluxo de tarefa B	fluxo de tarefa B utiliza o calendário A	Um erro especificando o relacionamento existente é exibido.

Tabela 59. Referential integrity check when deleting an object from the database (continuação)

Objeto a Ser Excluído	Referenciado pelo Objeto	Relacionamento	Regra de Exclusão
classe de estação de trabalho A	tarefa B	tarefa B está definida na classe de estação de trabalho A	Um erro especificando o relacionamento existente é exibido.
	fluxo de tarefa B	fluxo de tarefa B está definido na classe de estação de trabalho A	Um erro especificando o relacionamento existente é exibido.
	recurso B	recurso B está definido na classe de estação de trabalho A	Um erro especificando o relacionamento existente é exibido.
	arquivo B	arquivo B está definido na classe de estação de trabalho A	Um erro especificando o relacionamento existente é exibido.
recurso A	fluxo de tarefa B	dependência needs definida no fluxo de tarefa B	Um erro especificando o relacionamento existente é exibido.
	tarefa B no fluxo de tarefa B	dependência needs definida na tarefa B	Um erro especificando o relacionamento existente é exibido.
prompt A	fluxo de tarefa B	dependência prompt definida no fluxo de tarefa B	Um erro especificando o relacionamento existente é exibido.
	tarefa B no fluxo de tarefa B	dependência prompt definida na tarefa B	Um erro especificando o relacionamento existente é exibido.
variável A	fluxo de tarefa B	variável A é utilizada no fluxo de tarefa B em: • no texto de um prompt ad hoc • ou no nome de arquivo especificado em uma dependência de arquivo	variável A é excluída sem verificação
	tarefa B	variável A é utilizada no fluxo de tarefa B em: • no texto de um prompt ad hoc • ou no nome de arquivo especificado em uma dependência de arquivo • ou no valor especificado para streamlogon • ou no valor especificado para scriptname	variável A é excluída sem verificação
	prompt B	variável A é utilizada no texto do prompt B	variável A é excluída sem verificação
tabela da variável A	fluxo de tarefa B	tabela de variáveis A é referenciada no fluxo de tarefa B	tabela de variáveis A não é excluída
	tarefa B	tabela de variáveis A é referenciada na tarefa B	tabela de variáveis A não é excluída
	prompt B	tabela de variáveis A é referenciada no texto do prompt B	tabela de variáveis A não é excluída
fluxo de tarefa A	fluxo de tarefa B	fluxo de tarefa B segue o fluxo de tarefa A	O fluxo de tarefa A e a dependência follows no fluxo de tarefa B são excluídos.
	tarefa B em um fluxo de tarefa B	tarefa B segue o fluxo de tarefa A	O fluxo de tarefa A e a dependência follows na tarefa B são excluídos.
	regra de evento B	fluxo de tarefas A está na definição de ação da regra de evento B (e não utiliza substituição de variáveis)	Um erro especificando o relacionamento existente é exibido.

add

Inclui ou atualiza objetos de planejamento no banco de dados.

Autorização

Você deve ter o acesso ***add*** para incluir um novo objeto de planejamento. Se o objeto já existir no banco de dados, você deve ter:

- acesso ***modify*** para o objeto se o objeto não estiver bloqueado.
- acesso ***modify*** e ***unlock*** para o objeto se o objeto estiver bloqueado por outro usuário.

Sintaxe

{**add** | **a**} *filename* [**;unlock**]

Argumentos

nome do arquivo
: Especifica o nome do arquivo de texto que contém as definições de objetos. Para regras de eventos, *filename* especifica o nome do arquivo XML que contém as definições das regras de evento que você deseja incluir (consulte "Definição de Regra de Evento" na página 260 para referência de XML e consulte "O Editor do Composer" na página 278 para obter detalhes sobre como configurar um editor de XML).

;unlock
: Indica que as definições de objetos devem ser desbloqueadas se foram bloqueadas pelo mesmo usuário na mesma sessão. Se você não tiver bloqueado o objeto e utilizar a opção ;unlock, quando emitir o comando você receberá uma mensagem de erro e o objeto será substituído.

Comentários

O arquivo de texto é validado no cliente e, se correto, os objetos são inseridos no banco de dados no gerenciador de domínio principal. O **composer** transforma as definições de objetos em uma definição XML utilizada no servidor; caso contrário, o comando é interrompido e uma mensagem de erro é exibida. Isto não se aplica às definições de regras de eventos porque elas são fornecidas diretamente no formato XML.

Com o comando **add**, se um objeto já existir, será perguntado se você deseja ou não substituí-lo. Esse comportamento não afeta as definições de tarefas nos fluxos de tarefas e as definições de tarefas são atualizadas automaticamente sem solicitar qualquer mensagem. É possível utilizar a opção **unlock** para atualizar os objetos existentes anteriormente bloqueados utilizando somente um comando. Para todos os novos objetos inseridos, a opção é ignorada. Se você alterar o nome de um objeto, ele é interpretado pelo **composer** como o novo objeto e será inserido. Um comando **rename** é recomendado neste caso.

O comando add verifica as dependências de loop dentro dos fluxos de tarefas. Por exemplo, se `job1` seguir `job2` e `job2` seguir `job1`, haverá uma dependência de loop. Quando for localizada uma dependência de loop dentro de um fluxo de tarefas, um erro será exibido. O comando add não verifica as dependências de loop entre os fluxos de tarefas porque, dependendo da complexidade das atividades de planejamento, esta verificação poderá ser muito demorada e consumir muita CPU.

Nota: Para importar para o banco de dados os dados salvos nos arquivos simples criados utilizando a linha de comandos **composer** em outra instância do Tivoli Workload Scheduler versão V8.3 ou posterior, use o utilitário **datamigrate** no gerenciador de domínio principal em vez de utilizar o comando **composer** para gerenciar os arquivos simples extraídos. Para obter informações adicionais sobre o utilitário **datamigrate**, consulte "datamigrate" na página 503.

Exemplos

Para incluir os jobs a partir do arquivo `myjobs`, execute o seguinte comando:

```
add myjobs
```

Para incluir os fluxos de job a partir do arquivo `mysked`, execute o seguinte comando:

```
a mysked
```

Para incluir as estações de trabalho, as classes de estações de trabalho e os domínios a partir do arquivo `cpus.src`, execute o seguinte comando:

```
a cpus.src
```

Para incluir as definições de usuário a partir do arquivo `users_nt`, execute o seguinte comando:

```
a users_nt
```

Para incluir as definições de regras de eventos editadas em um arquivo denominado `newrules.xml`, execute:

```
a newrules.xml
```

Consulte também

A partir do Dynamic Workload Console, é possível executar as mesmas tarefas conforme descrito em:

no Guia do Usuário do Console do Dynamic Workload.

- Para incluir estações de trabalho, consulte

  o Guia do Usuário do Console do Dynamic Workload, seção Criando Cargas de Trabalho Distribuídas.
- Para incluir regras de eventos, consulte

  o Guia do Usuário do Console do Dynamic Workload, seção Criando uma Regra de Evento.
- Para incluir todos os outros objetos, consulte

  o Guia do Usuário do Console do Dynamic Workload, seção Projetando sua Carga de Trabalho.

authenticate

Alterna para as credenciais de outro usuário durante a execução do **composer**.

Autorização

Qualquer usuário autorizado a executar o **composer** está autorizado a emitir esse comando.

Sintaxe

{**authenticate** | **au**} [**username=***username* **password=***password*]

Argumentos

username=*username*
O nome do usuário para o qual deseja alternar.

password=*password*
A senha do usuário para a qual deseja alternar.

Comentários

Uma mensagem é exibida comunicando a falha ou o sucesso da autenticação. Esse comando é utilizado somente no modo interativo.

Exemplos

Para alternar para o usuário **tws_user1** com a senha **mypasswd1** a partir do programa de linha de comandos **composer**, execute o seguinte comando:

```
au username=tws_user1 password=mypasswd1
```

continue

Especifica que o próximo erro do comando será ignorado.

Autorização

Qualquer usuário autorizado a executar o composer está autorizado a emitir esse comando.

Sintaxe

{**continue** | **c**}

Comentários

Esse comando é útil quando vários comandos são digitados na linha de comandos ou redirecionados a partir de um arquivo. Ele orienta o **composer** a continuar a execução de comandos, mesmo que o próximo comando, depois de **continue**, resulte em erro. Esse comando não é necessário quando você digita comandos interativamente porque o **composer** não sairá com erro.

Exemplos

Para que o composer continue com o comando **print** se o comando **delete** resultar em erro, execute o seguinte comando:

```
composer "c&delete cpu=site4&print cpu=@"
```

delete

Exclui definições de objetos do banco de dados.

Autorização

Você deve ter acesso ***delete*** aos objetos que estão sendo excluídos.

Sintaxe

{**delete** | **de**}
{[**calendars** | **calendar** | **cal**=*calname*] |
[**domain** | **dom**=*domainame*] |
[**eventrule** | **erule** | **er**=*eventrulename*] |
[**parms** | **parm** | **vb**=[*tablename.*]*variablename*] |
[**prompts** | **prom**=*promptname*] |
[**resources** | **resource** | **res**=[*workstationame#*]*resourcename*] |
[**runcyclegroup** | **rcg**=*runcyclegroupname*] |
[**vartable** | **vt**=*tablename*] |
[**wat**=*workloadapplicationtemplatename*]
[**cpu**={*workstationame* [**;force**]
| *workstationclassname* [**;force**]|
domainame}]
[**workstation** | **ws**=*workstationame*] [**;force**] |

[**workstationclass** | **wscl**=*workstationclassname*] [**;force**] |
[**jobs** | **jobdefinition** | **jd**=[*workstationame*#]*jobname*] |
[**sched** | **jobstream** | **js**= [*workstationame*#]*jstreamname*
*[**valid from** date|**valid to** date |**valid in** date date]*
] |
[**users** | **user**=[*workstationame*#]*username*]}
[**;noask**]

Argumentos

calendars | calendar | cal
Se nenhum argumento vier em seguida, esse comando excluirá todas as definições de calendário.

Se o argumento *calname* vier em seguida, excluirá o calendário especificado. Caracteres curinga são permitidos.

domain | dom
Se nenhum argumento vier em seguida, esse comando excluirá todas as definições de domínio.

Se o argumento *domainname* vier em seguida, excluirá o domínio especificado. Caracteres curinga são permitidos.

eventrule | erule | er
Se nenhum argumento vier em seguida, esse comando excluirá todas as definições de regra de evento.

Se o argumento *eventrulename* vier em seguida, excluirá a regra de evento especificada. Caracteres curinga são permitidos.

parms | parm | vb
Se nenhum argumento vier em seguida, esse comando excluirá todas as definições de variáveis globais localizadas na tabela de variáveis padrão.

Se o argumento *tablename.variablename* vier em seguida, esse comando excluirá a variável *variablename* da tabela *tablename*. Se *tablename* for omitido, **composer** procurará a definição da variável na tabela de variáveis padrão. Caracteres curinga são permitidos em *tablename* e *variablename*. Por exemplo:

```
delete parms=@.@
```

Exclui todas as variáveis de todas as tabelas.

```
delete parms=@
```

Exclui todas as variáveis da tabela padrão.

```
delete parms=@.acct@
```

Exclui todas as variáveis cujo nome inicia com `acct` a partir de todas as tabelas existentes.

Lembre-se: Enquanto você exclui uma variável, a tabela de variáveis em que ela está contida é bloqueada. Isso indica que, enquanto a tabela está bloqueada, nenhum outro usuário pode executar nenhum outro comando de bloqueio nela ou nas variáveis que ela contém.

prompts | prom
Se nenhum argumento vier em seguida, esse comando excluirá todas as definições de prompt.

Se o argumento *promptname* vier em seguida, excluirá o prompt especificado. Caracteres curinga são permitidos.

resources | resource | res
Se nenhum argumento vier em seguida, esse comando excluirá todas as definições de recurso.

Se o argumento *workstationname#resourcename* vier em seguida, esse comando excluirá o recurso *resourcename* da estação de trabalho *workstationname* em que o recurso está definido. Se *workstationname* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationname* e *resourcename*.

runcyclegroup | rcg
Se nenhum argumento vier em seguida, exclui todas as definições do grupo do ciclo de execução.

Se o argumento *runcyclegroupname* vier em seguida, excluirá o grupo do ciclo de execução especificado. Caracteres curinga são permitidos.

vartable | vt
Se nenhum argumento vier em seguida, esse comando excluirá todas as definições de tabela de variáveis.

Se a tabela de variáveis *tablename* do argumento vier em seguida, excluirá a tabela de variáveis especificada. Caracteres curinga são permitidos.

wat Se nenhum argumento vier em seguida, excluirá todas as definições de modelo de aplicativo de carga de trabalho.

Se o argumento *workloadapplicationtemplate* vier em seguida, excluirá o modelo de aplicativo de carga de trabalho especificado. Caracteres curinga são permitidos.

cpu Exclui estações de trabalho, classes de estação de trabalho ou domínios.

estação de trabalho
O nome da estação de trabalho. Caracteres curinga são permitidos. Se o argumento **force** for especificado, a definição da estação de trabalho é removida do banco de dados Tivoli Workload Scheduler.

workstationclass
O nome da classe de estação de trabalho. Caracteres curinga são permitidos. Se o argumento **force** for especificado, a definição de classe de estação de trabalho é removida do banco de dados Tivoli Workload Scheduler.

domínio
O nome do domínio. Caracteres curinga são permitidos.

workstation | ws
Se nenhum argumento vier em seguida, esse comando excluirá todas as definições de estação de trabalho.

Se o argumento *workstationname* vier em seguida, excluirá a estação de trabalho especificada. Caracteres curinga são permitidos. Se o argumento **force** for especificado, a definição da estação de trabalho é removida do banco de dados Tivoli Workload Scheduler.

workstationclass | wscl
Se nenhum argumento vier em seguida, esse comando excluirá todas as definições de classe de estação de trabalho.

Se o argumento *workstationclassname* vier em seguida, excluirá a classe da estação de trabalho especificada. Caracteres curinga são permitidos. Se o argumento **force** for especificado, a definição da classe da estação de trabalho será removida do banco de dados Tivoli Workload Scheduler.

jobs | jobdefinition | jd
Se nenhum argumento vier em seguida, esse comando excluirá todas as definições de tarefa.

Se o argumento *workstationame#jobname* vier em seguida, ele excluirá a tarefa *jobname* da estação de trabalho *workstationame* em que a tarefa é executada. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *jobname*.

sched | jobstream | js
Se nenhum argumento vier em seguida, esse comando excluirá todas as definições de fluxo de tarefas.

Se o argumento *workstationame#jstreamname* vier em seguida, ele excluirá o fluxo de tarefas *jstreamname* da estação de trabalho *workstationame* em que o fluxo de tarefas está definido. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *jstreamname*.

válido a partir de
date Restringe a seleção de fluxos de tarefas que possuem uma data *valid from* igual ao valor indicado. O formato é *mm/dd/aaaa*.

válido até
date Restringe a seleção de fluxos de tarefas que possuem uma data *valid to* igual ao valor indicado. O formato é *mm/dd/aaaa*.

valid in
date date O espaço de tempo durante o qual o fluxo de tarefa pode ser executado. O formato é *mm/dd/aaaa - mm/dd/aaaa*. Uma das duas datas pode ser representada por @.

users | user
Se nenhum argumento vier em seguida, esse comando excluirá todas as definições de usuário.

Se o argumento *workstationame#username* vier em seguida, ele excluirá o usuário *username* da estação de trabalho *workstationame* em que o usuário está definido. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *username*. O campo de senha não é copiado por motivos de segurança.

;noask Especifica para não solicitar confirmação antes da tomada de ação em cada objeto qualificado.

Comentários

Se você utilizar caracteres curinga para especificar um conjunto de definições, o **composer** requererá confirmação antes de excluir cada definição correspondente. Uma confirmação será requerida antes de excluir cada definição correspondente se a opção **noask** não for especificada.

Para excluir um objeto, ele não deve estar bloqueado. Se alguns objetos correspondentes estiverem bloqueados durante o processamento de comando, uma

mensagem de erro com a lista desses objetos será mostrada para o usuário.

Exemplos

Para excluir o `job3` ativado na estação de trabalho `site3`, execute o seguinte comando:

```
delete jobs=site3#job3
```

Para excluir todas as estações de trabalho com nomes começando por `ux`, execute o seguinte comando:

```
de cpu=ux@
```

Para excluir todos os fluxos de job com nomes começando por `test` em todas as estações de trabalho, execute o seguinte comando:

```
de sched=@#test@
```

Para excluir todas as regras de eventos denominadas de `rulejs320` a `rulejs329`, execute o seguinte comando:

```
de erule=rulejs32?
```

Consulte também

A partir do Dynamic Workload Console, é possível executar as mesmas tarefas conforme descrito em:

no Guia do Usuário do Console do Dynamic Workload.

- Para visualizar, editar ou excluir estações de trabalho, consulte

  o Guia do Usuário do Console do Dynamic Workload, seção Editando Definições da Estação de Trabalho.
- Para visualizar, editar ou excluir regras de evento, consulte

  o Guia do Usuário do Console do Dynamic Workload, seção Editando uma Regra de Evento.
- Para visualizar, editar ou excluir todos os outros objetos, consulte

  o Guia do Usuário do Console do Dynamic Workload, seção Listando Definições de Objetos no Banco de Dados.

display

Exibe os detalhes de uma ou mais definições de objetos do mesmo tipo armazenados no banco de dados. Toda a definição do objeto é exibida.

Autorização

Você deve ter acesso ***display*** ao objeto que está sendo exibido. Se quiser utilizar a palavra-chave **full**, deve ter também o acesso ***display*** para as tarefas contidas na definição do fluxo de tarefa. Se você não tiver o acesso necessário, o composer não poderá localizar os objetos.

Sintaxe

{**display** | **di**}
{[**calendars** | **calendar** | **cal**=*calname*] |
[**eventrule** | **erule** | **er**=*eventrulename*] |
[**parms** | **parm** | **vb**=*variablename.*]*variablename*] |
[**vartable** | **vt**=*tablename*] |

[**prompts** | **prom**=*promptname*] |
[**resources** | **resource** | **res**=[*workstationname*#]*resourcename*] |
[**runcyclegroup** | **rcg**=*runcyclegroupname*] |
[**cpu**={*workstationname* | *workstationclassname* | *domainame*}]
[**workstation** | **ws**=*workstationname*] |
[**workstationclass** | **wscl**=*workstationclassname*] |
[**domain** | **dom**=*domainame*] |
[**jobs** | **jobdefinition** | **jd**=[*workstationname*#]*jobname*] |
[**sched** | **jobstream** | **js**= [*workstationname*#]*jstreamname*
*[**valid from** date|**valid to** date |**valid in** date date]*
*[;**full**]* |
[**users** | **user**=[*workstationname*#]*username*]}
[**;offline**]

Argumentos

calendars | calendar | cal

Se nenhum argumento vier em seguida, esse comando exibirá todas as definições de calendário.

Se o argumento *calname* vier em seguida, ele exibirá o calendário *calname*. Caracteres curinga são permitidos.

eventrule | erule | er

Se nenhum argumento vier em seguida, esse comando exibirá todas as definições de regra de evento.

Se o argumento *eventrulename* vier em seguida, ele exibirá a regra de evento *eventrulename*. Caracteres curinga são permitidos.

parms | parm | vb

Se nenhum argumento vier em seguida, esse comando exibirá todas as definições de variáveis globais localizadas na tabela de variáveis padrão.

Se o argumento *tablename.variablename* vier em seguida, esse comando exibirá a variável *variablename* da tabela especificada. Se a tabela de variáveis *tablename* for omitida, **composer** procurará a definição da variável na tabela de variáveis padrão. Caracteres curinga podem ser utilizados na tabela de variáveis *tablename* e na tabela *variablename*. Por exemplo:

```
display parms=@.@
```

Exibe todas as variáveis em todas as tabelas.

```
display parms=@
```

Exibe todas as variáveis da tabela padrão.

```
display parms=@.acct@
```

Exibe todas as variáveis cujo nome inicia com `acct` em todas as tabelas existentes.

vartable | vt

Se nenhum argumento vier em seguida, esse comando exibirá todas as definições de tabela de variáveis.

Se a tabela de variáveis *tablename* do argumento vier em seguida, esse comando exibirá a tabela de variáveis *tablename*. Caracteres curinga são permitidos.

prompts | prom

Se nenhum argumento vier em seguida, esse comando exibirá todas as definições de prompt.

Se o argumento *promptname* vier em seguida, ele exibirá o prompt *promptname*. Caracteres curinga são permitidos.

resources | resource | res
Se nenhum argumento vier em seguida, esse comando exibirá todas as definições de recurso.

Se o argumento *workstationame#resourcename* vier em seguida, esse comando exibirá o recurso *resourcename* da estação de trabalho *workstationame* em que o recurso está definido. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *resourcename*.

runcyclegroup | rcg
Se nenhum argumento vier em seguida, exibe todas as definições do grupo do ciclo de execução.

Se o argumento *runcyclegroupname* vier em seguida, exibe o grupo do ciclo de execução *runcyclegroupname*. Caracteres curinga são permitidos.

cpu Exibe estações de trabalho, classes de estação de trabalho ou domínios.

estação de trabalho
O nome da estação de trabalho. Caracteres curinga são permitidos.

workstationclass
O nome da classe de estação de trabalho. Caracteres curinga são permitidos.

domínio
O nome do domínio. Caracteres curinga são permitidos.

workstation | ws
Se nenhum argumento vier em seguida, esse comando exibirá todas as definições de estação de trabalho.

Se o argumento *workstationname* vier em seguida, ele exibirá a estação de trabalho *workstationname*. Caracteres curinga são permitidos.

domain | dom
Se nenhum argumento vier em seguida, esse comando exibirá todas as definições de domínio.

Se o argumento *domainname* vier em seguida, ele exibirá o domínio *domainname*. Caracteres curinga são permitidos.

workstationclass | wscl
Se nenhum argumento vier em seguida, esse comando exibirá todas as definições de classe de estação de trabalho.

Se o argumento *workstationclassname* vier em seguida, ele exibirá a classe de estação de trabalho *workstationclassname*. Caracteres curinga são permitidos.

jobs | jobdefinition | jd
Se nenhum argumento vier em seguida, esse comando exibirá todas as definições de tarefa.

Se o argumento *workstationame#jobname* vier em seguida, exibirá a tarefa *jobname* da estação de trabalho *workstationame* em que a tarefa é executada. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *jobname*.

sched | jobstream | js
Se nenhum argumento vier em seguida, esse comando exibirá todas as definições de fluxo de tarefas.

Se o argumento *workstationame#jstreamname* vier em seguida, ele exibirá o fluxo de tarefas *jstreamname* da estação de trabalho *workstationame* em que o fluxo de tarefas está definido. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *jstreamname*.

válido a partir de
date Restringe a seleção de fluxos de tarefas que possuem uma data *valid from* igual ao valor indicado. O formato é *mm/dd/aaaa*.

válido até
date Restringe a seleção de fluxos de tarefas que possuem uma data *valid to* igual ao valor indicado. O formato é *mm/dd/aaaa*.

valid in
date date O espaço de tempo durante o qual o fluxo de tarefa pode ser executado. O formato é *mm/dd/aaaa - mm/dd/aaaa*. Uma das duas datas pode ser representada por @.

full Exibe também todas as definições de tarefas contidas no fluxo de tarefas.

users | user
Se nenhum argumento vier em seguida, esse comando exibirá todas as definições de usuário.

Se o argumento *workstationame#username* vier em seguida, exibirá o usuário *username* da estação de trabalho *workstationame* em que o usuário está definido. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *username*.

;offline
Envia a saída do comando para o dispositivo de saída do **composer**. Para obter informações sobre esse dispositivo, consulte "Variáveis do UNIX" na página 278.

Resultados

O comando **display** retorna as seguintes informações sobre o objeto a ser exibido:

- uma linha de resumo contendo informações sobre o objeto selecionado
- a definição do objeto selecionado

Dependendo do valor configurado na variável local ***MAESTROCOLUMNS***, a linha de resumo mostra diferentes conjuntos de informações sobre o objeto selecionado.

A Tabela 60 mostra um exemplo da saída produzida com base no valor configurado para a variável ***MAESTROCOLUMNS***.

Tabela 60. Formatos de Saída para Exibir Objetos de Planejamento

Tipo do Objeto	Formato de saída se MAESTROCOLUMNS<120	Formato de Saída Se MAESTROCOLUMNS ≥ 120
Calendário	"CalendarName : UpdatedOn : UpdatedBy : LockedBy"	"CalendarName : UpdatedBy : UpdatedOn : LockedBy : LockedOn"
Domínio	"DomainName : ParentDomain : Master : UpdatedOn : LockedBy"	"DomainName : ParentDomain : Master : UpdatedBy : UpdatedOn : LockedBy : LockedOn"
Regra de Evento	"EventRuleName : Type : Draft : Status : UpdatedOn : LockedBy"	"EventRuleName : Type : Draft : Status : UpdatedOn : LockedBy : LockedOn"

Tabela 60. Formatos de Saída para Exibir Objetos de Planejamento (continuação)

Tipo do Objeto	Formato de saída se MAESTROCOLUMNS<120	Formato de Saída Se MAESTROCOLUMNS ≥ 120
Tarefa	"Workstation : JobDefinitionName : UpdatedOn : LockedBy"	"Workstation : JobDefinitionName : TaskType : UpdatedBy : LockedBy : LockedOn"
Fluxo de Tarefas	"Workstation : JobstreamName : Validfrom : UpdatedOn : LockedBy"	"Workstation : JobstreamName : Draft : ValidFrom : ValidTo : UpdatedBy : UpdatedOn : LockedBy : LockedOn"
Parâmetro	"VariableTableName : VariableName : UpdatedOn : LockedBy"	"VariableTableName : VariableName : UpdatedBy : UpdatedOn : LockedBy : LockedOn"
Aviso	"PromptName : UpdatedOn : LockedBy "	"PromptName : UpdatedBy : UpdatedOn : LockedBy : LockedOn"
Resource	"Workstation : ResourceName : Quantity : UpdatedOn : LockedBy "	"Workstation : ResourceName : Quantity : UpdatedBy : UpdatedOn : LockedBy : LockedOn"
Tabela de Variáveis	"VariableTableName : Default : UpdatedOn : LockedBy "	"VariableTableName : Default : UpdatedBy : UpdatedOn : LockedBy : LockedOn"
Usuário	"Workstation : UserName : UpdatedOn : LockedBy"	"UserName : Workstation : UpdatedBy : UpdatedOn : LockedBy : LockedOn"
Estação de Trabalho	"WorkstationName : Type : Domain : Ignored : UpdatedOn : LockedBy"	"WorkstationName : Type : Domain : OsType : Ignored : UpdatedBy : UpdatedOn : LockedBy : LockedOn"
Classe de Estação de Trabalho	"WorkstationClassName : Ignored : UpdatedOn : LockedBy"	"WorkstationClassName : Ignored : UpdatedBy : UpdatedOn : LockedBy : LockedOn"

Consulte “Saída Offline” na página 277 para obter informações adicionais sobre como configurar ***MAESTROCOLUMNS***.

Exemplos

Para exibir todos os calendários, execute o seguinte comando:

```
display calendars=@
```

esta é uma saída de amostra:

```
Calendar Name  Updated On Locked By
---------------  ----------  --------------------
HOLIDAYS   12/31/2005 tws83
HOLIDAYS
  01/01/2006 02/15/2006 05/31/2006


Calendar Name  Updated On Locked By
---------------  ----------  --------------------
MONTHEND   01/01/2006 -

MONTHEND
  "Month end dates 1st half 2006"
  01/31/2006 02/28/2006 03/31/2006 04/30/2006 05/31/2006 06/30/2006

Calendar Name  Updated On Locked By
---------------  ----------  --------------------
PAYDAYS   01/02/2006 -

PAYDAYS
  01/15/2006 02/15/2006 03/15/2006 04/15/2006 05/14/2006 06/15/2006
```

Para imprimir a saída do comando display de todos os fluxos de tarefa ativados na estação de trabalho `site2`, execute o seguinte comando:

```
di sched=site2#@;off-line
```

Consulte também

A partir do Dynamic Workload Console, é possível executar as mesmas tarefas conforme descrito em:

no Guia do Usuário do Console do Dynamic Workload.

- Para visualizar, editar ou excluir estações de trabalho, consulte

  o Guia do Usuário do Console do Dynamic Workload, seção Editando Definições da Estação de Trabalho.
- Para visualizar, editar ou excluir regras de evento, consulte

  o Guia do Usuário do Console do Dynamic Workload, seção Editando uma Regra de Evento.

- Para visualizar, editar ou excluir todos os outros objetos, consulte o Guia do Usuário do Console do Dynamic Workload, seção Listando Definições de Objetos no Banco de Dados.

edit

Edita um arquivo.

Autorização

Qualquer usuário autorizado a executar o composer está autorizado a emitir esse comando.

Sintaxe

{**edit** | **ed**} *filename*

Argumentos

nome do arquivo
O nome do arquivo a ser editado.

Comentários

Um editor é iniciado e o arquivo especificado é aberto para edição. Consulte "O Editor do Composer" na página 278 para obter informações adicionais.

Exemplos

Para abrir o arquivo `mytemp` para edição, execute o seguinte comando:

```
edit mytemp
```

Para abrir o arquivo `resfile` para edição, execute o seguinte comando:

```
ed resfile
```

exit

Sai do programa de linha de comandos do **composer**.

Autorização

Qualquer usuário autorizado a executar o composer está autorizado a emitir esse comando.

Sintaxe

{**exit** | **e**}

Comentários

Quando o programa de linha de comandos do **composer** está sendo executado no modo de ajuda, esse comando retorna o **composer** para o modo de entrada do comando.

Exemplos

Para sair do programa de linha de comandos do **composer**, execute o seguinte comando:

```
exit
```

ou:

```
J
```

extract

Cria um arquivo de texto que contém definições de objetos extraídas do banco de dados.

Autorização

Você deve ter o acesso ***display*** para os objetos que estão sendo copiados e, se quiser utilizar a palavra-chave **;lock**, também o acesso ***modify***. Se você não tiver o acesso necessário, o composer não poderá localizar os objetos.

Sintaxe

{**create** | **cr** | **extract** | **ext**} *filename* **from**
{[**calendars** | **calendar** | **cal**=*calname*] |
[**eventrule** | **erule** | **er**=*eventrulename*] |
[**parms** | **parm** | **vb**=[*tablename*.]*variablename*] |
[**vartable** | **vt**=*tablename*] |
[**prompts** | **prom**=*promptname*] |
[**resources** | **resource** | **res**=[*workstationname*#]*resourcename*] |
[**runcyclegroup** | **rcg**=*runcyclegroupname*] |
[**cpu**={*workstationname* | *workstationclassname* | *domainame*}] |
[**workstation** | **ws**=*workstationname*] |
[**workstationclass** | **wscl**=*workstationclassname*] |
[**domain** | **dom**=*domainame*] |
[**jobs** | **jobdefinition** | **jd**=[*workstationname*#]*jobname*] |
[**sched** | **jobstream** | **js**= [*workstationname*#]*jstreamname*
*[**valid from** date|**valid to** date |**valid in** date date]*
*[**;full**]*] |
[**users** | **user**=[*workstationame*#]*username*
[;password]]}
[**;lock**]

Argumentos

nome do arquivo
Especifica o nome do arquivo que contém as definições de objetos.

calendars | calendar | cal
Se nenhum argumento vier em seguida, esse comando copiará todas as definições do calendário no arquivo.

Se o argumento *calname* vier em seguida, ele copiará o calendário *calname* no arquivo. Caracteres curinga são permitidos.

eventrule | erule | er
Se nenhum argumento vier em seguida, esse comando copiará todas as definições de regra de evento no arquivo XML.

Se o argumento *eventrulename* vier em seguida, ele copiará a regra de evento *eventrulename* no arquivo. Caracteres curinga são permitidos.

parms | parm | vb
Se nenhum argumento vier em seguida, esse comando copiará todas as definições de variáveis globais localizadas na tabela de variáveis padrão no arquivo.

Se o argumento *tablename.variablename* vier em seguida, esse comando copiará a variável *variablename* da tabela de variáveis *tablename* especificada para o arquivo. Se a tabela de variáveis *tablename* for omitida, **composer** procurará a definição da variável na tabela de variáveis padrão. Caracteres curinga são permitidos na tabela de variáveis *tablename* e na variável *variablename*.

Por exemplo:

```
create parmfile from parms=@.@
```

Copia todas as variáveis de todas as tabelas.

```
create parmfile from parms=@
```

Copia todas as variáveis da tabela padrão.

```
create parmfile
from parms=@.acct@
```

Copia todas as variáveis cujo nome inicia com `acct` a partir de todas as tabelas existentes.

Lembre-se: Utilizar a opção **;lock** em uma variável bloqueia a tabela de variáveis que ela contém. Isso indica que, enquanto a tabela está bloqueada, nenhum outro usuário pode executar nenhum outro comando de bloqueio nela ou nas variáveis que ela contém.

vartable | vt
Se nenhum argumento vier em seguida, esse comando copiará todas as definições da tabela de variáveis no arquivo.

Se a tabela de variáveis *tablename* do argumento vier em seguida, esse comando copiará a tabela de variáveis *tablename* no arquivo. Caracteres curinga são permitidos.

prompts | prom
Se nenhum argumento vier em seguida, esse comando copiará todas as definições de prompt no arquivo.

Se o argumento *promptname* vier em seguida, ele copiará o prompt *promptname* no arquivo. Caracteres curinga são permitidos.

resources | resource | res
Se nenhum argumento vier em seguida, esse comando copiará todas as definições de recurso no arquivo.

Se o argumento *workstationame#resourcename* vier em seguida, copiará o recurso *resourcename* da estação de trabalho *workstationame* em que o recurso está definido no arquivo. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *resourcename*.

runcyclegroup | rcg
Se nenhum argumento vier em seguida, copia todas as definições do grupo de ciclo de execução no arquivo.

Se um argumento *runcyclegroupname* vier em seguida, copia o grupo de ciclo de execução *runcyclegroupname* no arquivo. Caracteres curinga são permitidos.

cpu Copia estações de trabalho, classes de estação de trabalho ou domínios para o arquivo.

estação de trabalho
O nome da estação de trabalho. Caracteres curinga são permitidos.

workstationclass
O nome da classe de estação de trabalho. Caracteres curinga são permitidos.

domínio
O nome do domínio. Caracteres curinga são permitidos.

workstation | ws
Se nenhum argumento vier em seguida, todas as definições de estação de trabalho serão copiadas no arquivo.

Se o argumento *workstationname* vier em seguida, copiará a estação de trabalho *workstationname* no arquivo. Caracteres curinga são permitidos.

domain | dom
Se nenhum argumento vier em seguida, todas as definições de domínio serão copiadas no arquivo.

Se o argumento *domainname* vier em seguida, o domínio *domainname* será copiado no arquivo. Caracteres curinga são permitidos.

workstationclass | wscl
Se nenhum argumento vier em seguida, todas as definições de classe de estação de trabalho serão copiadas no arquivo.

Se o argumento *workstationclassname* vier em seguida, copiará a classe de estação de trabalho *workstationclassname* no arquivo. Caracteres curinga são permitidos.

jobs | jobdefinition | jd
Se nenhum argumento vier em seguida, todas as definições de tarefa serão copiadas no arquivo.

Se o argumento *workstationame#jobname* vier em seguida, copiará a tarefa *jobname* da estação de trabalho *workstationame* em que a tarefa é executada no arquivo. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *jobname*.

sched | jobstream | js
Se nenhum argumento vier em seguida, todas as definições de fluxo de tarefas serão copiadas no arquivo.

Se o argumento *workstationame#jstreamname* vier em seguida, copiará o fluxo de tarefas *jstreamname* da estação de trabalho *workstationame* em que o fluxo de tarefas está definido no arquivo. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *jstreamname*.

válido a partir de
date Restringe a seleção de fluxos de tarefas que possuem uma data *valid from* igual ao valor indicado. O formato é *mm/dd/aaaa*.

válido até
date Restringe a seleção de fluxos de tarefas que possuem uma data *valid to* igual ao valor indicado. O formato é *mm/dd/aaaa*.

valid in
date date O espaço de tempo durante o qual o fluxo de tarefa pode ser executado. O formato é *mm/dd/aaaa - mm/dd/aaaa*. Uma das duas datas pode ser representada por @.

full Copia também todas as definições de tarefas contidas no fluxo de tarefas.

users | user
Se nenhum argumento vier em seguida, todas as definições de usuário serão copiadas no arquivo.

Se o argumento *workstationame#username* vier em seguida, copiará o usuário *username* da estação de trabalho *workstationame* em que o usuário está definido no arquivo. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *username*.

Se você não incluir a opção **;password**, a senha definida para o usuário será salva no arquivo de saída como um sequência de 10 asteriscos (*) e não poderá ser reutilizada.

Se você incluir a opção **;password**, a senha definida para o usuário será criptografada e salva no arquivo de saída. Ela pode, assim, ser reimportada e usada novamente.

;lock Especifica que o objeto selecionado deve ser mantido bloqueado.

Comentários

É possível utilizar este comando para criar um arquivo que contém definições de parâmetros a serem importadas no banco de dados de parâmetros definido localmente em uma estação de trabalho. Para obter informações adicionais sobre como importar definições de parâmetros localmente, consulte “parms” na página 520.

É possível chamar o comando com o nome antigo “create” ou o novo nome “extract”. Sem a opção **lock**, o bloqueio de banco de dados não é verificado e todos os objetos correspondentes são extraídos para o arquivo. Após criar um arquivo, é possível utilizar o comando **edit** para fazer alterações no arquivo e o comando **add** ou **replace** para incluir ou atualizar o banco de dados.

É possível especificar com a opção **lock** se os objetos que respondem aos critérios selecionados precisam permanecer bloqueados pelo usuário no banco de dados. Se o **composer**, durante a extração, localizar algum desses objetos já bloqueados por outra pessoa, esses objetos não serão inseridos no arquivo e uma mensagem para *stdout* será apresentada para cada objeto bloqueado.

Exemplos

Para criar um arquivo denominado `caltemp` contendo todos os calendários, execute o seguinte comando:

```
create caltemp from calendars=@
```

Para criar um arquivo denominado `stemp` que contenha todos os fluxos de tarefa definidos na estação de trabalho em que o **composer** é executado, execute o seguinte comando:

```
cr stemp from jobstream=@
```

Para criar um arquivo denominado `alljobs.txt` contendo todas as definições de tarefa, execute o seguinte comando:

```
extract alljobs.txt from jd=@#@
```

Para criar um arquivo denominado `allrules.xml` contendo todas as definições de regras de eventos, execute o seguinte comando:

```
ext allrules.xml from erule=@
```

Para criar um arquivo chamado `dbmainadm.txt` com a definição de usuário `princeps` da estação de trabalho `dbserv349`, incluindo a senha criptografada, execute:

```
composer extract
c:\dbmainadm.txt from user=dbserv349#princeps;password
```

O conteúdo do arquivo `dbmainadm.txt` será:

```
USERNAME princeps
  PASSWORD "ENCRYPT:EIu7PP+gvS8="
END
```

help

Exibe a ajuda online para um comando ou exibe a lista de comandos que podem ser emitidos a partir do **composer**. Não disponível no Windows.

Autorização

Qualquer usuário autorizado a executar o composer está autorizado a emitir esse comando.

Sintaxe

{**help** | **h**} {*command* | *keyword*}

Argumentos

comando

Especifica o nome de um comando do **composer** ou do sistema. Para comandos do **composer**, digite o nome completo do comando; abreviações e formas curtas não são suportadas.

keyword

Você também pode digitar as seguintes palavras-chave:

COMMANDS
Lista todos os comandos do composer.

RUNCONPOSER
Como executar o composer.

SETUPCOMPOSER
Descreve como configurar para usar o composer.

SPECIALCHAR
Descreve curingas, delimitadores e outros caracteres especiais que é possível usar.

Exemplos

Para exibir uma lista de todos os comandos do **composer**, execute o comando a seguir:

```
help commands
```

Para exibir informações sobre o comando **add**, execute o comando a seguir:

```
help add
```

Para exibir informações sobre caracteres especiais que é possível usar, execute o comando a seguir:

```
h specialchar
```

list

Lista ou imprime informações de resumo sobre os objetos definidos no banco de dados do Tivoli Workload Scheduler. List fornece a lista de nomes de objetos com seus atributos. Print envia a lista de nomes de objetos com seus atributos para o dispositivo ou o arquivo especificado na variável local ***MAESTROLP***. O comando print pode ser utilizado para enviar a saída para uma impressora local, se a variável ***MAESTROLP*** estiver definida adequadamente.

Autorização

Se a opção global ***enListSecChk*** estiver configurada como **yes** no gerenciador de domínio principal, para listar um objeto, você deve ter o acesso ***list*** ou ***list*** e ***display***.

Sintaxe

{**list** | **l**}
{[**calendars** | **calendar** | **cal**=*calname*] |
[**eventrule** | **erule** | **er**=*eventrulename*] |
[**parms** | **parm** | **vb**=[*tablename*.]*variablename*] |
[**vartable** | **vt**=*tablename*] |
[**prompts** | **prom**=*promptname*] |
[**resources** | **resource** | **res**=[*workstationname*#]*resourcename*] |
[**runcyclegroup** | **rcg**=*runcyclegroupname*] |
[**cpu**={*workstationname* | *workstationclassname* | *domainame*}]
[**workstation** | **ws**=*workstationname*] |
[**workstationclass** | **wscl**=*workstationclassname*] |
[**domain** | **dom**=*domainame*] |
[**jobs** | **jobdefinition** | **jd**=[*workstationname*#]*jobname*] |
[**sched** | **jobstream** | **js**= [*workstationname*#]*jstreamname*
*[**valid from** date*|
valid to** date* |valid in** date date]* |
[**users** | **user**=[*workstationname*#]*username*]}
[**;offline**]

Argumentos

calendars | calendar | cal

Se nenhum argumento vier em seguida, esse comando listará ou imprimirá todas as definições de calendário.

Se o argumento *calname* vier em seguida, ele listará ou imprimirá o calendário *calname*. Caracteres curinga são permitidos.

eventrule | erule | er
Se nenhum argumento vier em seguida, esse comando listará ou imprimirá todas as definições de regra de evento.

Se o argumento *eventrulename* vier em seguida, ele listará ou imprimirá a regra de evento *eventrulename*. Caracteres curinga são permitidos.

parms | parm | vb
Se nenhum argumento vier em seguida, esse comando listará ou imprimirá todas as definições de variáveis globais localizadas na tabela de variáveis padrão.

Se o argumento *tablename.variablename* vier em seguida, esse comando listará ou imprimirá a variável *variablename* da tabela *tablename*. Se *tablename* for omitido, **composer** procurará a definição da variável na tabela de variáveis padrão. Caracteres curinga podem ser utilizados em *tablename* e *variablename*. Por exemplo:

```
list parms=@.@
```

Lista todas as variáveis contidas em todas as tabelas.

```
list parms=@
```

Lista todas as variáveis da tabela padrão.

```
list parms=@.acct@
```

Lista todas as variáveis cujo nome inicia com `acct` em todas as tabelas existentes.

vartable | vt
Se nenhum argumento vier em seguida, esse comando listará ou imprimirá todas as definições de tabela de variáveis.

Se a tabela de variáveis *tablename* do argumento vier em seguida, esse comando listará ou imprimirá a tabela de variáveis *tablename*. Caracteres curinga são permitidos.

prompts | prom
Se nenhum argumento vier em seguida, esse comando listará ou imprimirá todas as definições de prompt.

Se o argumento *promptname* vier em seguida, ele listará ou imprimirá o prompt *promptname*. Caracteres curinga são permitidos.

resources | resource | res
Se nenhum argumento vier em seguida, esse comando listará ou imprimirá todas as definições de recurso.

Se o argumento *workstationname#resourcename* vier em seguida, esse comando listará ou imprimirá o recurso *resourcename* da estação de trabalho *workstationname* em que o recurso está definido. Se *workstationname* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationname* e *resourcename*.

runcyclegroup | rcg
Se nenhum argumento vier em seguida, lista ou imprime todos os grupos de ciclos de execução.

Se o argumento *runcyclegroupname* vier em seguida, lista ou imprime o grupo do ciclo de execução *runcyclegroupname*. Caracteres curinga são permitidos.

cpu Lista ou imprime estações de trabalho, classes de estação de trabalho ou domínios.

estação de trabalho
O nome da estação de trabalho. Caracteres curinga são permitidos.

workstationclass
O nome da classe de estação de trabalho. Caracteres curinga são permitidos.

domínio
O nome do domínio. Caracteres curinga são permitidos.

workstation | ws
Se nenhum argumento vier em seguida, esse comando listará ou imprimirá todas as definições de estação de trabalho.

Se o argumento *workstationname* vier em seguida, ele listará ou imprimirá a estação de trabalho *workstationname*. Caracteres curinga são permitidos.

domain | dom
Se nenhum argumento vier em seguida, esse comando listará ou imprimirá todas as definições de domínio.

Se o argumento *domainname* vier em seguida, ele listará ou imprimirá o domínio *domainname*. Caracteres curinga são permitidos.

workstationclass | wscl
Se nenhum argumento vier em seguida, esse comando listará ou imprimirá todas as definições de classe de estação de trabalho.

Se o argumento *workstationclassname* vier em seguida, ele listará ou imprimirá a classe de estação de trabalho *workstationclassname*. Caracteres curinga são permitidos.

jobs | jobdefinition | jd
Se nenhum argumento vier em seguida, esse comando listará ou imprimirá todas as definições de tarefa.

Se o argumento *workstationame#jobname* vier em seguida, ele listará ou imprimirá a tarefa *jobname* da estação de trabalho *workstationame* em que a tarefa é executada. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *jobname*.

sched | jobstream | js
Se nenhum argumento vier em seguida, esse comando listará ou imprimirá todas as definições de fluxo de tarefas.

Se o argumento *workstationame#jstreamname* vier em seguida, ele listará ou imprimirá o fluxo de tarefas *jstreamname* da estação de trabalho *workstationame* em que o fluxo de tarefas está definido. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *jstreamname*.

válido a partir de
date Restringe a seleção de fluxos de tarefas que possuem uma data *valid from* igual ao valor indicado. O formato é *mm/dd/aaaa*.

válido até
date Restringe a seleção de fluxos de tarefas que possuem uma data *valid to* igual ao valor indicado. O formato é *mm/dd/aaaa*.

valid in

date date O espaço de tempo durante o qual o fluxo de tarefa pode ser executado. O formato é *mm/dd/aaaa - mm/dd/aaaa*. Uma das duas datas pode ser representada por @.

users | user

Se nenhum argumento vier em seguida, esse comando listará ou imprimirá todas as definições de usuário.

Se o argumento *workstationame#username* vier em seguida, ele listará ou imprimirá o usuário *username* da estação de trabalho *workstationame* em que o usuário está definido. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *username*.

Nota: Se você estiver listando usuários do Windows no formato UPN `username@internet_domain` , insira o caractere de escape '\' antes do caractere '@' no valor `username@internet_domain` . Por exemplo, se estiver listando o usuário `administrator@bvt.com`, execute o seguinte comando:

```
list users=administrator\@bvt.com
```

;offline

Envia a saída do comando para o dispositivo de saída do **composer**. Para obter informações sobre esse dispositivo, consulte "Variáveis do UNIX" na página 278. O comando **list ..... ;offline** é equivalente ao comando **print**.

Resultados

List fornece a lista de nomes de objetos com seus atributos. Print envia a lista de nomes de objetos com seus atributos para o dispositivo ou o arquivo definido na variável local *MAESTROLP*. O comando print poderá ser utilizado para enviar a saída para uma impressora local, se a variável *MAESTROLP* estiver definida adequadamente. Certifique-se de que *MAESTROLP* é configurado em seu ambiente antes de executar o comando de impressão.

Dependendo do valor configurado na variável local ***MAESTROCOLUMNS***, os diferentes conjuntos de informações sobre o objeto selecionado podem ser mostrados.

A Tabela 61 mostra um exemplo da saída produzida para o valor configurado para a variável ***MAESTROCOLUMNS***.

Tabela 61. Formatos de Saída para Exibir Objetos de Planejamento

Tipo do Objeto	Formato de saída se MAESTROCOLUMNS<120	Formato de Saída Se MAESTROCOLUMNS ≥ 120
Calendário	"CalendarName : UpdatedOn : UpdatedBy : LockedBy"	"CalendarName : UpdatedBy : UpdatedOn : LockedBy : LockedOn"
Domínio	"DomainName : ParentDomain : Master : UpdatedOn : LockedBy"	"DomainName : ParentDomain : Master : UpdatedBy : UpdatedOn : LockedBy : LockedOn"
Regra de Evento	"EventRuleName : Type : Draft : Status : UpdatedOn : LockedBy"	"EventRuleName : Type : Draft : Status : UpdatedOn : LockedBy : LockedOn"
Tarefa	"Workstation : JobDefinitionName : UpdatedOn : LockedBy"	"Workstation : JobDefinitionName : TaskType : UpdatedBy : LockedBy : LockedOn"
Fluxo de Tarefas	"Workstation : JobstreamName : Validfrom : UpdatedOn : LockedBy"	"Workstation : JobstreamName : Draft : ValidFrom : ValidTo : UpdatedBy : UpdatedOn : LockedBy : LockedOn"
Parâmetro	"VariableTableName : VariableName : UpdatedOn : LockedBy"	"VariableTableName : VariableName : UpdatedBy : UpdatedOn : LockedBy : LockedOn"
Aviso	"PromptName : UpdatedOn : LockedBy "	"PromptName : UpdatedBy : UpdatedOn : LockedBy : LockedOn"
Resource	"Workstation : ResourceName : Quantity : UpdatedOn : LockedBy "	"Workstation : ResourceName : Quantity : UpdatedBy : UpdatedOn : LockedBy : LockedOn"

Tabela 61. Formatos de Saída para Exibir Objetos de Planejamento (continuação)

Tipo do Objeto	Formato de saída se MAESTROCOLUMNS<120	Formato de Saída Se MAESTROCOLUMNS ≥ 120
Grupo do Ciclo de Execução	"RunCycleGroupName : UpdatedOn : LockedBy "	"RunCycleGroupName : UpdatedBy : UpdatedOn : LockedBy : LockedOn"
Tabela de Variáveis	"VariableTableName : Default : UpdatedOn : LockedBy "	"VariableTableName : Default : UpdatedBy : UpdatedOn : LockedBy : LockedOn"
Usuário	"Workstation : UserName : UpdatedOn : LockedBy"	"UserName : Workstation : UpdatedBy : UpdatedOn : LockedBy : LockedOn"
Estação de Trabalho	"WorkstationName : Type : Domain : Ignored : UpdatedOn : LockedBy"	"WorkstationName : Type : Domain : OsType : Ignored : UpdatedBy : UpdatedOn : LockedBy : LockedOn"
Classe de Estação de Trabalho	"WorkstationClassName : Ignored : UpdatedOn : LockedBy"	"WorkstationClassName : Ignored : UpdatedBy : UpdatedOn : LockedBy : LockedOn"

Consulte “Saída Offline” na página 277 para obter informações adicionais sobre como configurar ***MAESTROLP*.**

Exemplos

- Para listar todos os calendários, execute o seguinte comando:

```
list calendars=@
```

esta é uma saída de amostra:

```
Calendar Name  Updated On Locked By
---------------  ----------  ---------------------
HOLIDAYS   03/02/2010
PAYDAYS   03/02/2010
HOLIDAYS   03/02/2010
  01/01/2010 02/15/2010 05/31/2010


Calendar Name  Updated On Locked By
---------------  ----------  ---------------------
MONTHEND   01/01/2010 -

MONTHEND
  "Month end dates 1st half 2010"
  01/31/2010 02/28/2010 03/31/2010 04/30/2010 05/31/2010 06/30/2010

Calendar Name  Updated On Locked By
---------------  ----------  ---------------------
PAYDAYS   01/02/2010 -

PAYDAYS
  01/15/2010 02/15/2010 03/15/2010 04/15/2010 05/14/2010 06/15/2010
```

- Para listar todas as suas regras de eventos definidas, execute o seguinte comando:

```
list er=@
```

Se `MAESTROCOLUMNS=80`, a saída será semelhante a isto:

```
Event Rule Name       Type  Draft Status Updated On Locked By
---------------  ---------  -----  ---------  ----------  -------------
EVENT-MULTIPLE1   filter   active 06/06/2009 -
EVENT-MULTIPLE2   filter   active 06/06/2009 -
EVENT-MULTIPLE3   filter   active 06/06/2009 -
M_SUCC_12_S    sequence Y inactive 06/07/2009 -
M_SUCC_12_S_A   filter   active 06/07/2009 -
M_SUCC_12_S_B   filter  Y inactive 06/07/2009 -
NEWEVENTRULE   filter   active 06/01/2009 administrator
```

Se `MAESTROCOLUMNS≥120`, a saída será semelhante a isto:

```
Event Rule Name       Type  Draft Status Updated On Locked By
--------------------   --------- ----- -------- ---------- ---------
EVENT-MULTIPLE1   filter   active 06/06/2009 -
EVENT-MULTIPLE2   filter   active 06/06/2009 -
EVENT-MULTIPLE3   filter   active 06/06/2009 -
M_SUCC_12_S    sequence Y inactive 06/07/2009 -
M_SUCC_12_S_A   filter   active 06/07/2009 -
M_SUCC_12_S_B   filter  Y inactive 06/07/2009 -
NEWEVENTRULE   filter   active 06/01/2009 administrator
```

- Para visualizar as propriedades da estação de trabalho NC1150691 do agent, execute o seguinte comando:

```
list ws=NC1150691
```

Uma saída semelhante à seguinte é exibida:

```
Workstation Name Type  Domain   Ignored Updated On Locked By
---------------  -------  ---------------  -------  ----------  ----------
NC1150691   agent -       03/31/2010 -

CPUNAME NC1150691
```

```
  DESCRIPTION "This workstation was automatically created at agent
               installation time."
  OS WNT
  NODE nc115069.romelab.it.ibm.com SECUREADDR 22114
  TIMEZONE GMT+1
  FOR MAESTRO HOST NC115069_DWB
    TYPE AGENT
    PROTOCOL HTTPS
END
```

- Para visualizar as propriedades da estações de trabalho do conjunto POOL_A, incluindo todos seus membros, execute o comando a seguir:

```
list ws=POOL_A
```

  Uma saída semelhante à seguinte é exibida:

```
Workstation Name Type  Domain   Ignored Updated On Locked By
---------------  ------- ---------------  -------  ----------  ----------------
POOL_A   pool  -       03/31/2010 -

CPUNAME POOL_A
  DESCRIPTION "This is a manually created pool"
  VARTABLE TABLE1
  OS OTHER
  TIMEZONE America/Argentina/Buenos_Aires
  FOR MAESTRO HOST NC115069_DWB
    TYPE POOL
  MEMBERS
    NC1150691
    NC1150692
END
```

Consulte também

A partir do Dynamic Workload Console, é possível executar as mesmas tarefas conforme descrito em:

no Guia do Usuário do Console do Dynamic Workload.

- Para visualizar, editar ou excluir estações de trabalho, consulte

  o Guia do Usuário do Console do Dynamic Workload, seção Editando Definições da Estação de Trabalho.
- Para visualizar, editar ou excluir regras de evento, consulte

  o Guia do Usuário do Console do Dynamic Workload, seção Editando uma Regra de Evento.
- Para visualizar, editar ou excluir todos os outros objetos, consulte

  o Guia do Usuário do Console do Dynamic Workload, seção Listando Definições de Objetos no Banco de Dados.

lock

Bloqueia o acesso às definições dos objetos de planejamento no banco de dados.

Autorização

Você deve ter acesso ***modify*** para o objeto.

Sintaxe

{**lock** | **lo**}
{[**calendars** | **calendar** | **cal**=*calname*] |
[**eventrule** | **erule** | **er**=*eventrulename*]
[**parms** | **parm** | **vb**=[*tablename.*]*variablename*] |
[**vartable** | **vt**=*tablename*] |
[**prompts** | **prom**=*promptname*] |

[**resources** | **resource** | **res**=[*workstationname*#]*resourcename*] |
[**runcyclegroup** | **rcg**=*runcyclegroupname*] |
[**cpu**={*workstationname* | *workstationclassname* | *domainame*}]
[**workstation** | **ws**=*workstationname*] |
[**workstationclass** | **wscl**=*workstationclassname*] |
[**domain** | **dom**=*domainame*] |
[**jobs** | **jobdefinition** | **jd**=[*workstationname*#]*jobname*] |
[**sched** | **jobstream** | **js**= [*workstationname*#]*jstreamname*
*[**valid from** date* | ***valid to** date* | ***valid in** date date]*] |
[**users** | **user**=[*workstationname*#]*username*]}

Argumentos

calendários
Bloqueia todas as definições de calendário.

calendars | calendar | cal
Se nenhum argumento vier em seguida, esse comando bloqueará todas as definições de calendário.

Se o argumento *calname* vier em seguida, ele bloqueará o calendário *calname*. Caracteres curinga são permitidos.

eventrule | erule | er
Se nenhum argumento vier em seguida, esse comando bloqueará todas as definições de regra de evento.

Se o argumento *eventrulename* vier em seguida, ele bloqueará a regra de evento *eventrulename*. Caracteres curinga são permitidos.

parms | parm | vb
Se nenhum argumento vier em seguida, esse comando bloqueará a tabela de variáveis padrão inteira.

Se o argumento *tablename.variablename* vier em seguida, esse comando bloqueará a tabela inteira que contém a variável *variablename*. Se *tablename* for omitido, **composer** bloqueará a tabela de variáveis padrão inteira.

Nota: Quando você bloqueia uma variável, a tabela de variáveis inteira em que ela está contida é bloqueada. Isso indica que, enquanto a tabela está bloqueada, nenhum outro usuário pode executar nenhum outro comando de bloqueio nela.
Caracteres curinga podem ser utilizados em *tablename* e *variablename*. Por exemplo:

```
lock parms=@.@
```

Bloqueia todas as variáveis em todas as tabelas. Como resultado, todas as tabelas de variáveis são bloqueadas.

```
lock parms=@
```

Bloqueia todas as variáveis na tabela padrão. Como resultado, a tabela de variáveis é bloqueada.

```
lock parms=@.acct@
```

Bloqueia todas as variáveis cujo nome inicia com `acct` em todas as tabelas existentes. Como resultado, todas as tabelas de variáveis que contêm pelo menos uma variável denominada dessa forma são bloqueadas.

vartable | vt
Se nenhum argumento vier em seguida, esse comando bloqueará todas as definições de tabela de variáveis.

Se a tabela de variáveis *tablename* do argumento vier em seguida, esse comando bloqueará a tabela de variáveis *tablename*. Caracteres curinga são permitidos.

prompts | prom
Se nenhum argumento vier em seguida, esse comando bloqueará todas as definições de prompt.

Se o argumento *promptname* vier em seguida, ele bloqueará o prompt *promptname*. Caracteres curinga são permitidos.

resources | resource | res
Se nenhum argumento vier em seguida, esse comando bloqueará todas as definições de recurso.

Se o argumento *workstationname#resourcename* vier em seguida, esse comando bloqueará o recurso *resourcename* da estação de trabalho *workstationname* em que o recurso está definido. Se *workstationname* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationname* e *resourcename*.

runcyclegroup | rcg
Se nenhum argumento vier em seguida, bloqueia todas as definições de grupo de ciclo de execução.

Se o argumento *runcyclegroupname* vier em seguida, bloqueia o grupo de ciclo de execução *runcyclegroupname*. Caracteres curinga são permitidos.

cpu Bloqueia estações de trabalho, classes de estação de trabalho ou domínios.

estação de trabalho
O nome da estação de trabalho. Caracteres curinga são permitidos.

workstationclass
O nome da classe de estação de trabalho. Caracteres curinga são permitidos.

domínio
O nome do domínio. Caracteres curinga são permitidos.

workstation | ws
Se nenhum argumento vier em seguida, esse comando bloqueará todas as definições de estação de trabalho.

Se o argumento *workstationname* vier em seguida, ele bloqueará a estação de trabalho *workstationname*. Caracteres curinga são permitidos.

domain | dom
Se nenhum argumento vier em seguida, esse comando bloqueará todas as definições de domínio.

Se o argumento *domainname* vier em seguida, ele bloqueará o domínio *domainname*. Caracteres curinga são permitidos.

workstationclass | wscl
Se nenhum argumento vier em seguida, esse comando bloqueará todas as definições de classe de estação de trabalho.

Se o argumento *workstationclassname* vier em seguida, ele bloqueará a classe de estação de trabalho *workstationclassname*. Caracteres curinga são permitidos.

jobs | jobdefinition | jd
Se nenhum argumento vier em seguida, esse comando bloqueará todas as definições de tarefa.

Se o argumento *workstationame#jobname* vier em seguida, ele bloqueará a tarefa *jobname* da estação de trabalho *workstationame* em que a tarefa é executada. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *jobname*.

sched | jobstream | js
Se nenhum argumento vier em seguida, esse comando bloqueará todas as definições de fluxo de tarefas.

Se o argumento *workstationame#jstreamname* vier em seguida, ele bloqueará o fluxo de tarefas *jstreamname* da estação de trabalho *workstationame* em que o fluxo de tarefas está definido. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *jstreamname*.

válido a partir de
date Restringe a seleção de fluxos de tarefas que possuem uma data *valid from* igual ao valor indicado. O formato é *mm/dd/aaaa*.

válido até
date Restringe a seleção de fluxos de tarefas que possuem uma data *valid to* igual ao valor indicado. O formato é *mm/dd/aaaa*.

valid in
date date O espaço de tempo durante o qual o fluxo de tarefa pode ser executado. O formato é *mm/dd/aaaa - mm/dd/aaaa*. Uma das duas datas pode ser representada por @.

users | user
Se nenhum argumento vier em seguida, esse comando bloqueará todas as definições de usuário.

Se o argumento *workstationame#username* vier em seguida, ele bloqueará o usuário *username* da estação de trabalho *workstationame* em que o usuário está definido. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *username*.

Comentários

Objetos são bloqueados para assegurar que as definições no banco de dados não sejam sobrescritas por diferentes usuários que estão acessando simultaneamente os mesmos objetos.

Com esse comando, o usuário adquire bloqueios de objetos de banco de dados explicitamente. Quando um usuário tiver um objeto bloqueado, qualquer outro usuário terá acesso de leitura até o objeto ser liberado ou explicitamente desbloqueado pelo administrador. Se um usuário tentar bloquear um objeto que já está bloqueado por outra pessoa (outro usuário), uma mensagem de erro é retornada.

Bloqueios em objetos de banco de dados são adquiridos pelo usuário utilizando `username` e `session`, em que `session` é uma cadeia que pode ser configurada na variável de ambiente **TWS_SESSION** que identifica essa sessão de trabalho do usuário específica.

Isto significa que, em uma máquina, o identificador **TWS_SESSION** é diferente para:

- um usuário conectado em dois shells diferentes ao programa de linha de comandos **composer**.
- um usuário conectado, desconectado e, em seguida, conectado novamente à linha de comandos **composer** a partir do mesmo shell.

Se nenhum valor for designado a **TWS_SESSION**, o valor padrão que identifica a sessão será configurado conforme a seguir:

- Se estiver utilizando **composer** no modo em lote, o valor padrão será o *username* utilizado pelo usuário ao se conectar ao gerenciador de domínio principal.
- Se estiver utilizando o **composer** no modo interativo, o valor padrão corresponderá a uma cadeia alfanumérica criada automaticamente pelo produto.

Nota: No banco de dados, o *username* do usuário que está bloqueando uma definição de objeto é salvo em maiúsculas.

Exemplos

Para bloquear o calendário denominado `Holidays`, execute o comando:

```
lock calendar=HOLIDAYS
```

Consulte também

No Dynamic Workload Console, os objetos são bloqueados automaticamente contanto que você ou outro usuário os tenha aberto utilizando o botão **Editar**. Os objetos não serão bloqueados se você ou outro usuário os abriu com **Visualizar**.

modify

Modifica ou adiciona objetos de planejamento. Ao modificar os objetos, o comando **modify** extrairá somente os objetos que poderão ser bloqueados pelo usuário atual.

Autorização

Você deve ter o acesso **add** se incluir um novo objeto de planejamento. Se o objeto já existir no banco de dados, você deverá ter acesso **modify** ao objeto, caso contrário, o composer não poderá localizados os objetos.

Sintaxe

{**modify** | **m**}
{[**calendars** | **calendar** | **cal**=*calname*] |
[**eventrule** | **erule** | **er**=*eventrulename*] |
[**parms** | **parm** | **vb**=[*tablename*.]*variablename*] |
[**vartable** | **vt**=*tablename*] |
[**prompts** | **prom**=*promptname*] |
[**resources** | **resource** | **res**=[*workstationname*#]*resourcename*] |
[**runcyclegroup** | **rcg**=*runcyclegroupname*] |
[**cpu**={*workstationname* | *workstationclassname* | *domainame*}]
[**workstation** | **ws**=*workstationname*] |

[**workstationclass** | **wscl**=*workstationclassname*] |
[**domain** | **dom**=*domainame*] |
[**jobs** | **jobdefinition** | **jd**=[*workstationame*#]*jobname*] |
[**sched** | **jobstream** | **js**= [*workstationame*#]*jstreamname*
*[**valid from** date | **valid to** date | **valid in** date date]*
*[**;full**]*] |
[**users** | **user**=[*workstationame*#]*username*]}

Argumentos

calendars | calendar | cal

Se nenhum argumento vier em seguida, esse comando modificará todas as definições de calendário.

Se o argumento *calname* vier em seguida, ele modificará o calendário *calname*. Caracteres curinga são permitidos.

eventrule | erule | er

Se nenhum argumento vier em seguida, esse comando modificará todas as definições de regra de evento.

Se o argumento *eventrulename* vier em seguida, ele modificará a regra de evento *eventrulename*. Caracteres curinga são permitidos.

parms | parm | vb

Se nenhum argumento vier em seguida, esse comando modificará todas as definições de variáveis globais localizadas na tabela de variáveis padrão.

Se o argumento *tablename.variablename* vier em seguida, esse comando modificará a variável especificada da tabela *tablename*. Se *tablename* for omitido, **composer** procurará a definição da variável *variablename* na tabela de variáveis padrão. Caracteres curinga podem ser utilizados em *tablename* e *variablename*. Por exemplo:

```
modify parms=@.@
```

Modifica todas as variáveis contidas em todas as tabelas.

```
modify parms=@
```

Modifica todas as variáveis da tabela padrão.

```
modify parms=@.acct@
```

Modifica todas as variáveis cujo nome inicia com `acct` em todas as tabelas existentes.

Lembre-se: A ação de modificar ou incluir uma variável bloqueia a tabela de variáveis que ela contém. Isso indica que, enquanto a tabela está bloqueada, nenhum outro usuário pode executar nenhum outro comando de bloqueio nela ou nas variáveis que ela contém.

vartable | vt

Se nenhum argumento vier em seguida, esse comando modificará todas as definições de tabela de variáveis.

Se a tabela de variáveis *tablename* do argumento vier em seguida, esse comando modificará a tabela de variáveis *tablename*. Caracteres curinga são permitidos.

prompts | prom

Se nenhum argumento vier em seguida, esse comando modificará todas as definições de prompt.

Se o argumento *promptname* vier em seguida, ele modificará o prompt *promptname*. Caracteres curinga são permitidos.

resources | resource | res
Se nenhum argumento vier em seguida, esse comando modificará todas as definições de recurso.

Se o argumento *workstationname#resourcename* vier em seguida, esse comando modificará o recurso *resourcename* da estação de trabalho *workstationname* em que o recurso está definido. Se *workstationname* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationname* e *resourcename*.

runcyclegroup | rcg
Se nenhum argumento vier em seguida, modifica todas as definições do grupo do ciclo de execução.

Se o argumento *runcyclegroupname* vier em seguida, modifica o grupo do ciclo de execução *runcyclegroupname*. Caracteres curinga são permitidos.

cpu Modifica estações de trabalho, classes de estação de trabalho ou domínios.

estação de trabalho
O nome da estação de trabalho. Caracteres curinga são permitidos.

workstationclass
O nome da classe de estação de trabalho. Caracteres curinga são permitidos.

domínio
O nome do domínio. Caracteres curinga são permitidos.

workstation | ws
Se nenhum argumento vier em seguida, esse comando modificará todas as definições de estação de trabalho.

Se o argumento *workstationname* vier em seguida, ele modificará a estação de trabalho *workstationname*. Caracteres curinga são permitidos.

domain | dom
Se nenhum argumento vier em seguida, esse comando modificará todas as definições de domínio.

Se o argumento *domainname* vier em seguida, ele modificará o domínio *domainname*. Caracteres curinga são permitidos.

workstationclass | wscl
Se nenhum argumento vier em seguida, esse comando modificará todas as definições de classe de estação de trabalho.

Se o argumento *workstationclassname* vier em seguida, ele modificará a classe de estação de trabalho *workstationclassname*. Caracteres curinga são permitidos.

jobs | jobdefinition | jd
Se nenhum argumento vier em seguida, esse comando modificará todas as definições de tarefa.

Se o argumento *workstationame#jobname* vier em seguida, ele modificará a tarefa *jobname* da estação de trabalho *workstationame* em que a tarefa é executada. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *jobname*.

sched | jobstream | js
Se nenhum argumento vier em seguida, esse comando modificará todas as definições de fluxo de tarefas.

Se o argumento *workstationame#jstreamname* vier em seguida, ele modificará o fluxo de tarefas *jstreamname* da estação de trabalho *workstationame* em que o fluxo de tarefas está definido. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *jstreamname*.

válido a partir de
date Restringe a seleção de fluxos de tarefas que possuem uma data *valid from* igual ao valor indicado. O formato é *mm/dd/aaaa*.

válido até
date Restringe a seleção de fluxos de tarefas que possuem uma data *valid to* igual ao valor indicado. O formato é *mm/dd/aaaa*.

valid in
date date O espaço de tempo durante o qual o fluxo de tarefa pode ser executado. O formato é *mm/dd/aaaa - mm/dd/aaaa*. Uma das duas datas pode ser representada por @.

full Modifica todas as definições de tarefas contidas no fluxo de tarefas.

users | user
Se nenhum argumento vier em seguida, esse comando modificará todas as definições de usuário.

Se o argumento *workstationame#username* vier em seguida, modificará o usuário *username* da estação de trabalho *workstationame* em que o usuário está definido. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *username*.

Comentários

O comando **modify** executa a seguinte seqüência de ações:

1. Bloqueia os objetos no banco de dados.
2. Copia a definição de objetos para um arquivo temporário.
3. Edita o arquivo.
4. Modifica a definição contida no arquivo temporário no banco de dados.
5. Se o comando **modify** falhar em um subconjunto dos objetos selecionados, o **composer** perguntará *"deseja editar novamente?"* e o arquivo salvo antes é reaberto para edição e as próximas etapas da seqüência são repetidas.
6. Desbloqueia os objetos no banco de dados.

Definições de regras de eventos são abertas com um editor de XML (consulte "Definição de Regra de Evento" na página 260 para referência XML e consulte "O Editor do Composer" na página 278 para obter detalhes sobre como configurar um editor de XML).

Se você modificar com o mesmo comando **modify** dois ou mais objetos vinculados por qualquer relacionamento, por exemplo, uma tarefa sucessora e sua tarefa predecessora, então, pode ser relevante para o resultado bem-sucedido do comando **modify** a ordem na qual os objetos são listados no arquivo temporário. Isto acontece porque o comando **modify** lê em seqüência os objetos contidos no

arquivo temporário; portanto, se o objeto de referência for exibido antes do objeto que está sendo referenciado, o comando modify poderá falhar no objeto de referência.

Por exemplo, se o comando:

```
modify FTA1#@PROVA
```

produzir o seguinte arquivo temporário:

```
SCHEDULE FTA1#PROVA VALIDFROM 08/31/2005
MATCHING SAMEDAY
:
FTA2#MY-JOB
 FOLLOWS FTA1#COPYOFPROVA.MY-JOB06
END

SCHEDULE FTA1#COPYOFPROVA VALIDFROM 08/31/2005
MATCHING SAMEDAY
:
FTA1#MY-JOBO6
END
```

e você alterar o nome da tarefa do predecessor de `FTA1#MY-JOB06` para `FTA1#MY-JOB05` em ambos os fluxos de tarefas, `FTA1#PROVA` e `FTA1#COPYOFPROVA`, o comando **modify**:

1. Primeiro tentará alterar a definição do fluxo de tarefas `FTA1#PROVA` e falhará porque ele localizará uma dependência follows de uma tarefa `FTA1#MY-JOB05` que ainda é desconhecida.
2. Em seguida, tentará alterar a definição de `FTA1#COPYOFPROVA` e será bem-sucedido.

Na segunda vez, execute **modify** para alterar a tarefa do predecessor de `FTA1#MY-JOB06` para `FTA1#MY-JOB05` no fluxo de tarefas `FTA1#PROVA`, o comando será desempenhado com êxito pois a tarefa do predecessor `FTA1#MY-JOB05` agora existirá no banco de dados.

Se o fluxo de tarefa `FTA1#COPYOFPROVA` tivesse sido listado no arquivo temporário antes de `FTA1#PROVA`, o comando **modify** teria sido executado com êxito na primeira vez porque o nome da tarefa do predecessor teria sido modificado antes de alterar a definição de dependência na tarefa sucessora.

Para definições do usuário, se o campo de senha mantiver o valor `"*******"` quando você sair do editor, a senha antiga será retida. Para especificar uma senha nula, utilize duas aspas duplas consecutivas ('').

O comando modify verifica as dependências de loop dentro dos fluxos de tarefas. Por exemplo, se `job1` seguir `job2` e `job2` seguir `job1`, haverá uma dependência de loop. Quando for localizada uma dependência de loop dentro de um fluxo de tarefas, um erro será exibido. O comando modify não verifica dependências de loop entre fluxos de tarefas porque, dependendo da complexidade das atividades de planejamento, esta verificação poderá ser muito demorada e consumir muita CPU.

Exemplos

Para modificar todos os calendários, execute o seguinte comando:

```
modify calendars=@
```

Para modificar o fluxo de tarefas `sked9` ativado na estação de trabalho `site1`, execute o seguinte comando:

```
m sched=site1#sked9
```

Para modificar todas as regras de eventos que incluem uma ação com a tarefa DPJOB10, execute:

```
mod er=@;filter job=DPJOB10
```

Consulte também

A partir do Dynamic Workload Console, é possível executar as mesmas tarefas conforme descrito em:

no Guia do Usuário do Console do Dynamic Workload.

- Para visualizar, editar ou excluir estações de trabalho, consulte

  o Guia do Usuário do Console do Dynamic Workload, seção Editando Definições da Estação de Trabalho.
- Para visualizar, editar ou excluir regras de evento, consulte

  o Guia do Usuário do Console do Dynamic Workload, seção Editando uma Regra de Evento.
- Para visualizar, editar ou excluir todos os outros objetos, consulte

  o Guia do Usuário do Console do Dynamic Workload, seção Listando Definições de Objetos no Banco de Dados.

new

Inclui uma nova definição de objeto de planejamento no banco de dados.

Autorização

Você deve ter o acesso ***add*** se incluir um novo objeto de planejamento. Se o objeto já existir no banco de dados, você deve ter o acesso ***modify*** para o objeto.

Sintaxe

new
[calendar |
domain |
eventrule |
job |
jobstream |
parameter |
prompt |
resource |
runcyclegroup |
user |
vartable |
workstation |
workstation_class]

Argumentos

O objeto que você deseja definir: um calendário, um domínio, uma regra de evento, uma tarefa, um fluxo de tarefas, uma variável, um prompt, um recurso, um usuário, uma tabela de variáveis, uma estação de trabalho ou uma classe de estação de trabalho.

Comentários

O comando abre um modelo predefinido que ajuda a editar a definição de objeto e a inclui no banco de dados quando você a salva.

Os modelos de objetos estão localizados na subpasta `templates` no diretório de instalação do Tivoli Workload Scheduler. Eles podem ser customizados para se ajustarem às suas preferências.

Definições de regras de eventos são abertas com um editor de XML (consulte "Definição de Regra de Evento" na página 260 para referência XML e consulte "O Editor do Composer" na página 278 para obter detalhes sobre como configurar um editor de XML).

Enquanto você cria uma variável, a tabela de variáveis de destino é bloqueada. Isso indica que, enquanto a tabela está bloqueada, nenhum outro usuário pode executar nenhum outro comando de bloqueio nela.

Exemplos

Para criar uma nova definição de usuário, execute:

```
new user
```

Para criar uma nova definição de prompt, execute:

```
new prompt
```

Para criar uma nova definição de regra de evento, execute:

```
new erule
```

Para criar uma nova definição de tabela de variáveis, execute:

```
new vartable
```

Para criar uma nova definição de variável, execute:

```
new parameter
```

Consulte também

A partir do Dynamic Workload Console, é possível executar as mesmas tarefas conforme descrito em:

no Guia do Usuário do Console do Dynamic Workload.

- Para criar estações de trabalho, consulte

  o Guia do Usuário do Console do Dynamic Workload, seção Criando Cargas de Trabalho Distribuídas.
- Para criar regras de eventos, consulte

  o Guia do Usuário do Console do Dynamic Workload, seção Criando uma Regra de Evento.
- Para criar todos os outros objetos, consulte

  o Guia do Usuário do Console do Dynamic Workload, seção Projetando sua Carga de Trabalho.

print

Esse é um sinônimo do comando **`list`**. Consulte "list" na página 305 para obter detalhes.

redo

Edita e executa o comando anterior novamente.

Nota: Se o comando anterior for **authenticate**, **redo** não exibe a senha especificada.

Autorização

Qualquer usuário autorizado a executar o composer está autorizado a emitir esse comando.

Sintaxe

{**redo** | **red**}

Contexto

Quando você executa o comando **redo**, o composer exibe o comando anterior para que possa ser editado e executado novamente. Utilize a barra de espaços para mover o cursor sob o caractere a ser modificado e digite as diretrizes a seguir.

Diretrizes

d[*dir***]** Exclui o caractere acima de **d**. Pode vir seguido por outras diretrizes.

i*text* Insere texto antes do caractere acima de **i**.

r*text* Substitui um ou mais caracteres por texto, começando com o caractere acima de **r**. A substituição estará implícita se nenhuma outra diretriz for digitada.

>*text* Anexa texto no final da linha.

>d[*dir* | *text***]**
Exclui caracteres no final da linha. Pode vir seguido por outra diretriz ou texto.

>r*text* Substitui caracteres no final da linha por texto.

Exemplos de Diretrizes

ddd Exclui os três caracteres acima de **d**s.

iabc Insere **abc** antes do caractere acima de **i**.

rabc Substitui os três caracteres, começando pelo caractere acima de **r**, por **abc**.

abc Substitui os três caracteres acima de **abc** por **abc**.

d diabc
Exclui o caractere acima do primeiro**d**, pula um caractere, exclui o caractere acima do segundo**d** e insere **abc** em seu lugar.

>abc Anexa **abc** no final da linha.

>ddabc
Exclui os dois últimos caracteres na linha e insere **abc** em seu lugar.

>rabc Substitui os três últimos caracteres na linha por **abc**.

Exemplos

Para inserir um caractere, execute o seguinte comando:

```
redo
dislay site1#sa@
   ip
display site1#sa@
```

Para substituir três caracteres, por exemplo, alterar `site` para `serv` substituindo`ite` por `erv`, execute o seguinte comando:

```
redo
display site1#sa@
         rerv
display serv1#sa@
```

rename

Renomeia um objeto de planejamento já existente no banco de dados. O novo nome não deve identificar um objeto já definido no banco de dados.

Autorização

Você deve ter o acesso ***delete*** para o objeto com o antigo nome e o acesso ***add*** para o objeto com o novo nome.

Sintaxe

{**rename** | **rn**}
{**calendars** | **calendar** | **cal** |
parms | **parm** | **vb** |
vartable | **vt** |
prompts | **prom** |
resorces | **resource** | **res** |
runcyclegroup | **rcg** |
workstation | **ws** |
workstationclass | **wscl** |
domain | **dom** |
jobs | **jobdefinition** | **jd** |
jobsched | **jb** |
eventrule | **erule** | **er**
sched | **jobstream** | **js**
|
users | **user** }
old_object_identifier new_object_identifier

Argumentos

old_object_identifier
Especifica a chave externa antiga que identifica o objeto de planejamento, por exemplo, o nome do calendário`cal1` como identificador para um objeto calendário definido para ser renomeado.

new_object_identifier
Especifica a nova chave externa para identificar o objeto de planejamento, por exemplo, o nome do calendário`cal2` como novo identificador a ser designado para o objeto calendário anteriormente denominado`cal1`.

No que se refere às tarefas, aos fluxos de tarefas, aos recursos e aos usuários, tanto***old_object_identifier*** quanto***new_object_identifier*** possuem os seguintes formatos:

[*workstationname***#**]*jobname*
O comando aplica-se a essa definição de tarefa. Esse formato é utilizado com a chave **jobs | jobdefinition | jd**.

[*workstationname***#**]*jstreamname*
O comando aplica-se a todas as versões desse fluxo de tarefas. Esse formato é utilizado com a chave **sched | jobstream | js**.

[*workstationname***#**]*jstreamname*+**valid from** *date*
O comando aplica-se somente a essa versão desse fluxo de tarefas. Esse formato é utilizado com a chave **sched | jobstream | js**.

[*workstationname***#**]*jstreamname.jobname*
O comando aplica-se a essa instância de tarefa definida nesse fluxo de tarefas. Consulte a palavra-chave **js** na sintaxe de "Definição de fluxo de tarefas" na página 216 para obter detalhes adicionais. Esse formato é utilizado com a chave **jobsched | jb**.

[*workstationname***#**]*resourcename*
O comando aplica-se a essa definição de recurso. Esse formato é utilizado como a chave **resources | resource | res**.

[*workstationname***#**][*domain*\]*username*
O comando se aplica a esta definição de usuário. Esse formato é utilizado com a chave **users | user**.

Para o que envolve as variáveis (parâmetros globais):

old_object_identifier
Deve ser especificado no formato *tablename.variablename*. Se *tablename* for omitido, composer procurará a variável na tabela de variáveis padrão.

new_object_identifier
Deve ser especificado no formato *variablename*. A inclusão do nome da tabela aqui gera um erro.

Comentários

Para ser renomeado, o objeto deve ser desbloqueado ou bloqueado pelo usuário que emite o comando rename.

A tabela de variáveis que contém a variável é bloqueada, enquanto a variável é renomeada. Isso indica que, enquanto a tabela está bloqueada, nenhum outro usuário pode executar nenhum outro comando de bloqueio nela.

Se um objeto denominado conforme especificado no campo *old_object_identifier* não existir no banco de dados, uma mensagem de erro é exibida.

O uso de curingas não é permitido com este comando.

Quando***workstationname*** não for especificado para os objetos que possuem o nome da estação de trabalho como parte de seus identificadores de objetos (por exemplo, definições de tarefas ou de fluxos de tarefas), o planejador utilizará um dos seguintes para ***workstationname*** :

- A estação de trabalho padrão especificada no arquivo *localopts*
- O gerenciador de domínio principal se o programa da linha de comandos **composer** estiver em execução em um nó fora da rede do Tivoli Workload Scheduler. Nesse caso, na verdade, a estação de trabalho padrão configurada no arquivo *localopts* é ogerenciador de domínio principal.

O comando **rename** é utilizado para designar novos nomes aos objetos já existentes no banco de dados. O novo nome designado a um objeto se torna imediatamente efetivo no banco de dados, enquanto ele se torna efetivo no plano após o script **JnextPlan** ser executado novamente. Isso pode levar a incongruências ao enviar tarefas ad hoc antes de gerar novamente o plano de produção.

Exemplos

Para renomear o objeto domínioDOMAIN1 para DOMAIN2, execute o seguinte comando:

```
rename dom=DOMAIN1 DOMAIN2
```

Para renomear o fluxo de tarefas LABJST1 para LABJST2 na estação de trabalho CPU1, execute o seguinte comando:

```
rename js=CPU1#LABJST1 CPU1#LABJST2
```

Para renomear a variável ACCTOLD (definida na tabela ACCTAB) para ACCTNEW, execute o seguinte comando:

```
rename
parm=ACCTAB.ACCTOLD ACCTNEW
```

Consulte também

A partir do Dynamic Workload Console, é possível executar as mesmas tarefas conforme descrito em:

no Guia do Usuário do Console do Dynamic Workload.

- Para visualizar, editar ou excluir estações de trabalho, consulte

  o Guia do Usuário do Console do Dynamic Workload, seção Editando Definições da Estação de Trabalho.
- Para visualizar, editar ou excluir regras de evento, consulte

  o Guia do Usuário do Console do Dynamic Workload, seção Editando uma Regra de Evento.
- Para visualizar, editar ou excluir todos os outros objetos, consulte

  o Guia do Usuário do Console do Dynamic Workload, seção Listando Definições de Objetos no Banco de Dados.

replace

Substitui as definições de objetos de planejamento no banco de dados.

Autorização

Você deve ter o acesso ***add*** se incluir um novo objeto de planejamento. Se o objeto já existir no banco de dados, você deve ter:

- acesso ***modify*** para o objeto se o objeto não estiver bloqueado.
- acessos ***modify*** e ***unlock*** para o objeto se quiser utilizar a opção **;unlock** com objetos bloqueados por outros usuários.

Sintaxe

{**replace** | **rep**} *filename* [**;unlock**]

Argumentos

nome do arquivo
Especifica o nome de um arquivo que contém as definições de objeto a serem substituídas. O arquivo pode conter todos os tipos de definição de objetos de planejamento.

destravar
Atualiza objetos existentes anteriormente bloqueados e desbloqueia-os. Um erro é exibido se os objetos não estiverem anteriormente bloqueados. Para todos os novos objetos inseridos, essa opção, se especificada, é ignorada.

Comentários

O comando **replace** é semelhante ao comando **add**, exceto por não haver nenhum prompt de confirmação para substituir objetos existentes. Para obter informações adicionais, consulte "add" na página 287.

O comando replace verifica as dependências de loop dentro dos fluxos de tarefas. Por exemplo, se `job1` seguir `job2` e `job2` seguir `job1`, haverá uma dependência de loop. Quando for localizada uma dependência de loop dentro de um fluxo de tarefas, um erro será exibido. O comando replace não verifica as dependências de loop entre os fluxos de tarefas porque, dependendo da complexidade das atividades de planejamento, esta verificação pode ser muito demorada e consumir muita CPU.

Exemplos

Para substituir os jobs a partir do arquivo `myjobs`, execute o seguinte comando:

```
replace myjobs
```

Para substituir todos os recursos por aqueles contidos no arquivo `myres`, execute o seguinte comando:

```
rep myres
```

Você deseja alterar algumas definições de regras de eventos existentes no banco de dados. Você também deseja incluir algumas definições novas. Utilize este comando da seguinte maneira:

1. Você grava as definições inteiras em um arquivo XML nomeado como `2Q07rules.xml` .
2. Você executa:

   ```
   rep 2Q07rules.xml
   ```

system command

Executa um comando do sistema.

Sintaxe

[**:** | **!**] *system-command*

Argumentos

system-command
Especifica qualquer comando de sistema válido. O prefixo de dois-pontos (:) ou ponto de exclamação (!) é necessário apenas quando o comando é grafado da mesma forma que um comando do composer.

Exemplos

Para executar um comando **ps** no UNIX, execute o seguinte comando:

```
ps -ef
```

Para executar um comando **dir** no Windows, execute o seguinte comando:

```
dir \bin
```

unlock

Libere os bloqueios de acesso nos objetos de planejamento definidos no banco de dados. Por padrão, para desbloquear um objeto, o objeto deve ter sido bloqueado utilizando-se o mesmo usuário e sessão.

Autorização

Você deve ter o acesso ***unlock*** para desbloquear objetos bloqueados por outros usuários.

Sintaxe

{**unlock** | **u**}
{[**calendars** | **calendar** | **cal**=*calname*] |
[**eventrule** | **erule** | **er**=*eventrulename*] |
[**parms** | **parm** | **vb**=[*tablename*.]*variablename*] |
[**vartable** | **vt**=*tablename*] |
[**prompts** | **prom**=*promptname*] |
[**resources** | **resource** | **res**=[*workstationname*#]*resourcename*] |
[**runcyclegroup** | **rcg**=*runcyclegroupname*] |
[**cpu**={*workstationname* | *workstationclassname* | *domainame*}]
[**workstation** | **ws**=*workstationname*] |
[**workstationclass** | **wscl**=*workstationclassname*] |
[**domain** | **dom**=*domainame*] |
[**jobs** | **jobdefinition** | **jd**=[*workstationname*#]*jobname*] |
[**sched** | **jobstream** | **js**= [*workstationname*#]*jstreamname*
*[**valid from** date* | ***valid to** date* | ***valid in** date date*]] |
[**users** | **user**=[*workstationname*#]*username*]}
[**;forced**]

Argumentos

calendars | calendar | cal
Se nenhum argumento vier em seguida, esse comando desbloqueará todas as definições de calendário.

Se o argumento *calname* vier em seguida, ele desbloqueará o calendário*calname*. Caracteres curinga são permitidos.

eventrule | erule | er
Se nenhum argumento vier em seguida, esse comando desbloqueará todas as definições de regra de evento.

Se o argumento *eventrulename* vier em seguida, ele desbloqueará a regra de evento*eventrulename*. Caracteres curinga são permitidos.

parms | parm | vb
Se nenhum argumento vier em seguida, esse comando desbloqueará a tabela de variáveis padrão.

Se o argumento *tablename.variablename* vier em seguida, esse comando desbloqueará a tabela inteira que contém a variável *variablename*. Se *tablename* for omitido, ele desbloqueará a tabela de variáveis padrão. Caracteres curinga podem ser utilizados em *tablename* e *variablename*. Por exemplo:

```
unlock parms=@.@
```

Desbloqueia todas as tabelas.

```
unlock parms=@
```

Desbloqueia a tabela padrão.

```
unlock parms=@.acct@
```

Desbloqueia todas as tabelas que contêm as variáveis cujo nome inicia com `acct`.

```
unlock parms=acct@
```

Desbloqueia a tabela padrão.

Lembre-se: A ação em uma variável única desbloqueia a tabela de variáveis que ela contém.

vartable | vt
Se nenhum argumento vier em seguida, esse comando desbloqueará todas as definições de tabela de variáveis.

Se a tabela de variáveis *tablename* do argumento vier em seguida, esse comando desbloqueará a tabela de variáveis *tablename*. Caracteres curinga são permitidos.

prompts | prom
Se nenhum argumento vier em seguida, esse comando desbloqueará todas as definições de prompt.

Se o argumento *promptname* vier em seguida, ele desbloqueará o prompt *promptname*. Caracteres curinga são permitidos.

resources | resource | res
Se nenhum argumento vier em seguida, esse comando desbloqueará todas as definições de recurso.

Se o argumento *workstationname#resourcename* vier em seguida, esse comando desbloqueará o recurso*resourcename* da estação de trabalho*workstationname* em que o recurso está definido. Se *workstationname* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationname* e *resourcename*.

runcyclegroup | rcg
Se nenhum argumento vier em seguida, desbloqueia todas as definições do grupo do ciclo de execução.

Se o argumento *runcyclegroupname* vier em seguida, desbloqueia o grupo do ciclo de execução *runcyclegroupname*. Caracteres curinga são permitidos.

cpu Desbloqueia estações de trabalho, classes de estação de trabalho ou domínios.

estação de trabalho
O nome da estação de trabalho. Caracteres curinga são permitidos.

workstationclass
O nome da classe de estação de trabalho. Caracteres curinga são permitidos.

domínio
O nome do domínio. Caracteres curinga são permitidos.

workstation | ws
Se nenhum argumento vier em seguida, esse comando desbloqueará todas as definições de estação de trabalho.

Se o argumento *workstationname* vier em seguida, ele desbloqueará a estação de trabalho*workstationname*. Caracteres curinga são permitidos.

domain | dom
Se nenhum argumento vier em seguida, esse comando desbloqueará todas as definições de domínio.

Se o argumento *domainname* vier em seguida, ele desbloqueará o domínio*domainname*. Caracteres curinga são permitidos.

workstationclass | wscl
Se nenhum argumento vier em seguida, esse comando desbloqueará todas as definições de classe de estação de trabalho.

Se o argumento *workstationclassname* vier em seguida, ele desbloqueará a classe de estação de trabalho*workstationclassname*. Caracteres curinga são permitidos.

jobs | jobdefinition | jd
Se nenhum argumento vier em seguida, esse comando desbloqueará todas as definições de tarefa.

Se o argumento *workstationame#jobname* vier em seguida, desbloqueará a tarefa *jobname* da estação de trabalho*workstationame* em que a tarefa é executada. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *jobname*.

sched | jobstream | js
Se nenhum argumento vier em seguida, esse comando desbloqueará todas as definições de fluxo de tarefas.

Se o argumento *workstationame#jstreamname* vier em seguida, desbloqueará o fluxo de tarefas *jstreamname* da estação de trabalho*workstationame* em que o fluxo de tarefas está definido. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *jstreamname*.

válido a partir de
date Restringe a seleção de fluxos de tarefas que possuem uma data *valid from* igual ao valor indicado. O formato é *mm/dd/aaaa*.

válido até
date Restringe a seleção de fluxos de tarefas que possuem uma data *valid to* igual ao valor indicado. O formato é *mm/dd/aaaa*.

valid in
date date O espaço de tempo durante o qual o fluxo de tarefa pode ser executado. O formato é *mm/dd/aaaa - mm/dd/aaaa*. Uma das duas datas pode ser representada por @.

users | user
Se nenhum argumento vier em seguida, esse comando desbloqueará todas as definições de usuário.

Se o argumento *workstationame#username* vier em seguida, desbloqueará o usuário *username* da estação de trabalho*workstationame* em que o usuário está definido. Se *workstationame* for omitido, o padrão será a estação de trabalho em que o **composer** está em execução. Caracteres curinga são permitidos para *workstationame* e *username*.

forced Se especificado, permite ao usuário que bloqueou o objeto desbloqueá-lo, independentemente da sessão.

Se esta opção for utilizada pelo *superuser* , o comando **unlock** poderá operar independentemente do usuário e da sessão utilizada para bloquear o objeto.

Comentários

Se um usuário, diferente do *superuser*, tentar desbloquear um objeto que está bloqueado por outro usuário, uma mensagem de erro é retornada.

Exemplos

Par desbloquear a definição de tarefa `JOBDEF1`, execute o seguinte comando:

```
unlock jd=@#JOBDEF1
```

Para desbloquear a definição da regra de evento`ERJS21`, execute o seguinte comando:

```
unlock erule=ERJS21
```

Consulte também

A partir do Dynamic Workload Console, é possível executar as mesmas tarefas conforme descrito em:

no Guia do Usuário do Console do Dynamic Workload.

- Para visualizar, editar ou excluir estações de trabalho, consulte
  o Guia do Usuário do Console do Dynamic Workload, seção Editando Definições da Estação de Trabalho.
- Para visualizar, editar ou excluir regras de evento, consulte
  o Guia do Usuário do Console do Dynamic Workload, seção Editando uma Regra de Evento.
- Para visualizar, editar ou excluir todos os outros objetos, consulte
  o Guia do Usuário do Console do Dynamic Workload, seção Listando Definições de Objetos no Banco de Dados.

validate

Desempenha a validação das definições de objetos contidas em um arquivo do usuário.

Autorização

Você não precisa de nenhuma autorização específica para objetos para executar esse comando.

Sintaxe

{**validate** | **val**} *filename* [**;syntax**]

Argumentos

nome do arquivo
Especifica o nome de um arquivo que contém calendários, estações de trabalho, classes de estação de trabalho, domínios, tarefas, parâmetros, prompts, recursos, fluxos de tarefas, regras de eventos ou tabelas de variáveis. Para definições de regras de eventos, o arquivo deve estar na linguagem XML. Consulte "Definição de Regra de Evento" na página 260 para obter detalhes sobre como gravar definições de regras de eventos.

sintaxe
Verifica se há erros de sintaxe no arquivo.

Comentários

A saída do comando **validate** pode ser redirecionada para um arquivo como o seguinte:

```
composer "validate filename" > outfile
```

Para incluir mensagens de erro no arquivo de saída, utilize o seguinte:

```
composer "validate filename" > outfile 2>&1
```

Exemplos

Para verificar a sintaxe de um arquivo contendo definições de estação de trabalho, execute o seguinte comando:

```
validate mycpus;syntax
```

version

Exibe o banner do programa de linha de comandos do **composer**.

Autorização

Qualquer usuário autorizado a executar o composer está autorizado a emitir esse comando.

Sintaxe

{**version** | **v**}

Exemplos

Para exibir o banner do programa de linha de comandos **composer**, execute o seguinte comando:

```
version
```

ou:

```
v
```

Capítulo 10. Gerenciando o aplicativos de carga de trabalho

Os Aplicativos de carga de trabalho podem ser criados e, em seguida, exportados para que possam ser compartilhados com outros ambientes do Tivoli Workload Scheduler. No novo ambiente, o aplicativo de carga de trabalho pode ser subsequentemente atualizado, substituído ou excluído.

O ciclo de vida de um aplicativo de carga de trabalho começa com a definição do modelo do aplicativo de carga de trabalho. O modelo é então exportado e, antes de ser implementado no ambiente de destino, é necessária alguma customização manual.

O processo de exportação produz um arquivo compactado contendo três arquivos:

*aplicativo de carga de trabalhoname*_**Definitions.UTF8.xml**
: Um arquivo em formato XML contendo uma definição de todos os objetos referenciados no aplicativo de carga de trabalho. As definições são implementadas no ambiente de destino para preencher o banco de dados de destino com os mesmos objetos existentes no ambiente de origem. Não edite esse arquivo.

*aplicativo de carga de trabalhoname*__**Mapping.UTF8.properties**
: Um arquivo de mapeamento de destino que o usuário modifica, substituindo os nomes dos objetos no ambiente de origem pelos nomes que os objetos terão no ambiente de destino.

*aplicativo de carga de trabalhoname*_**SourceEnv_reference.txt**
: Um arquivo contendo informações de referência sobre as estações de trabalho usadas no aplicativo de carga de trabalho e outras informações que podem ser úteis para mapear corretamente o ambiente de origem para o ambiente de destino.

Para usar o aplicativo de carga de trabalho em um novo ambiente, modifique o arquivo de mapeamento para refletir o ambiente de destino usando as informações fornecidas no arquivo de referência e, em seguida, execute uma operação de importação. A operação de importação é executada transmitindo o arquivo de mapeamento e o arquivo de definição como entrada para o comando wappman.

O comando wappman pode ser usado para importar, substituir, listar, exibir e excluir um aplicativo de carga de trabalho. Consulte “Comando wappman” na página 335 para obter o uso e a sintaxe completos da linha de comandos para executar estas ações em um aplicativo de carga de trabalho e quaisquer considerações especiais a serem observadas.

Resolvendo o Arquivo de Mapeamento

O arquivo de mapeamento produzido pelo processo de exportação de um aplicativo de carga de trabalho contém uma lista de elementos, alguns dos quais são dependentes da topologia do ambiente no qual ele é usado. Esses elementos precisam ser customizados para refletir o ambiente de destino.

Durante o processo de exportação, os objetos contidos no aplicativo de carga de trabalho são extraídos para o arquivo de definições com a mesma definição que eles possuem no ambiente de origem. O arquivo de definições pode conter uma definição de objeto completa ou, em alguns casos, apenas um nome ou referência

© Copyright IBM Corp. 1999, 2014

ao objeto extraído. Referências simples e não uma definição de objeto completa são extraídas para esses objetos que precisam ser mapeados para um objeto já presente no ambiente de destino. Para alguns objetos extraídos por referência, a definição de objeto é gravada no arquivo de mapeamento, que requer customização manual para mapear os objetos no ambiente de origem do Tivoli Workload Scheduler para o ambiente no qual o aplicativo de carga de trabalho será implementado.

O arquivo de mapeamento pode ser visualizado e editado com um editor de texto. Ele é organizado em seções e contém comentários para ajudar na designação dos valores corretos para os elementos.

A tabela a seguir descreve todos os objetos que podem estar contidos em um aplicativo de carga de trabalho ou que são referidos por outro elemento no aplicativo de carga de trabalho e como o processo de exportação os gerencia.

Tabela 62. Objetos extraídos durante o processo de exportação

Tipo de Objeto	O que é exportado para o arquivo de definições?	O que requer customização no arquivo de mapeamento?	O que acontece durante a importação?
Fluxo de Tarefas	Definição de objeto	Estação de Trabalho Nome do Fluxo de Tarefas Alias de Tarefa	O fluxo de tarefas é criado no banco de dados. **Restrição:** Se o fluxo de tarefas tiver uma dependência de um fluxo de tarefas ou de uma tarefa externa aoaplicativo de carga de trabalho, o arquivo de mapeamento conterá uma referência ao nome do fluxo de tarefas ou da tarefa externa e da definição da estação de trabalho relativa, no entanto, o arquivo de definições não conterá a definição do fluxo de tarefas ou da tarefa. O nome no arquivo de mapeamento deve ser mapeado para um fluxo de tarefas ou tarefa existente no ambiente de destino para a importação bem-sucedida do aplicativo de carga de trabalho.
Tarefa	Definição de objeto	Estação de Trabalho Nome da tarefa Afinidade Variáveis de Senha localizadas no JSDL	A tarefa é criada no banco de dados. **Restrição:** Se o fluxo de tarefas tiver uma dependência de uma tarefa que está definida em um fluxo de tarefas externo ao aplicativo de carga de trabalho, o arquivo de mapeamento conterá uma referência aos objetos a seguir: o nome do fluxo de tarefas externo, a tarefa definida no fluxo de tarefas externo, as definições da estação de trabalho do fluxo de tarefa e as definições da estação de trabalho da tarefa. No entanto, o arquivo de definição não contém a definição de tarefa. O nome no arquivo de mapeamento deve ser mapeado para um fluxo de tarefas ou tarefa existente no ambiente de destino para a importação bem-sucedida do aplicativo de carga de trabalho. Relacionamentos de afinidade fazem as tarefas executarem na mesma estação de trabalho. A estação de trabalho na qual a primeira tarefa é executada é escolhida dinamicamente e a tarefa ou tarefas de afinidade são executadas na mesma estação de trabalho. As tarefas devem pertencer ao mesmo fluxo de tarefas. Quando uma tarefa com uma afinidade é exportada, o nome da tarefa é incluído no arquivo de mapeamento. As variáveis no JSDL usam o formato ${password:ws#user}. Apenas as estações de trabalho são representadas genericamente. O campo de usuário é copiado como está para o ambiente de destino. As variáveis devem ser usadas para nomes de usuário.
Tabelas de variáveis de ciclo de execução Tabelas de variáveis de fluxo de tarefas	Definição de objeto	Nome da tabela Nome da variável	A tabela de variáveis é criada no banco de dados. As tabelas de variáveis da estação de trabalho e padrão são extraídas por referência e gravadas no arquivo de mapeamento. O valor associado à variável pode ser modificado, mas não os nomes de variáveis. Evite associar a tabela de variáveis padrão a fluxos de tarefas e ciclos de execução.
Usuário	Nada. Nem uma definição de objeto nem uma referência são feitas no arquivo de referência.		O usuário deve existir no ambiente de destino. As variáveis devem ser usadas para referir-se aos usuários para tornar oaplicativo de carga de trabalho mais flexível para reutilização.
Calendário	Definição de objeto	Nome do calendário	O calendário é criado no banco de dados
Ciclo de Execução	Definição de objeto	Nome do ciclo de execução	O ciclo de execução é criado no banco de dados
Grupo do Ciclo de Execução	Definição de objeto	Nome do grupo do ciclo de execução	O grupo do ciclo de execução é criado no banco de dados
Dependências da Interligação de Redes	A tarefa ou o fluxo de tarefas referenciado não é exportado porque pertence a um mecanismo diferente.	Nome da estação de trabalho do agente de rede que manipula as seguintes dependências entre a rede local e a rede remota.	A dependência de interligação de redes é incluída na tarefa ou no fluxo de tarefas.

Tabela 62. Objetos extraídos durante o processo de exportação (continuação)

Tipo de Objeto	O que é exportado para o arquivo de definições?	O que requer customização no arquivo de mapeamento?	O que acontece durante a importação?
Dependências externas	A tarefa ou fluxo de tarefas referenciado será exportado apenas se pertencer ao modelo do aplicativo de carga de trabalho.	Se a tarefa ou o fluxo de tarefas referenciado não pertencer ao modelo do aplicativo de carga de trabalho, designe um nome à tarefa ou ao fluxo de tarefas que corresponde a uma tarefa ou fluxo de tarefas já existente no ambiente de destino.	Dependência externa incluída na tarefa ou no fluxo de tarefas
Recursos	Definição de objeto	Estação de Trabalho Nome do recurso	Os recursos são criados no banco de dados
Prompt global	Definição de objeto	Variáveis usadas na definição	Os prompts globais são criados no banco de dados. Variáveis podem ser utilizadas. Como eles são resolvidos usando o valor padrão, a variável usada em um prompt global pode ser mapeada para uma variável no ambiente de destino.
Estações de Trabalho	Somente referência	Per.	Não importado. As estações de trabalho são extraídas no arquivo de definição como uma referência. Uma definição não é importada porque as estações de trabalho já estão definidas no ambiente de destino, no entanto, seus nomes precisam ser mapeados.
Classe de Estação de Trabalho	Somente referência	Per.	Não importado. As classes da estação de trabalho são extraídas no arquivo de definição como uma referência. Uma definição não é importada porque as classes da estação de trabalho já estão definidas no ambiente de destino, no entanto, seus nomes precisam ser mapeados.
Variável	Definição de objeto	Valor	Importada. As variáveis são usadas em vários locais em uma definição de fluxo de tarefas. Uma referência é incluída no arquivo de definições.

Exemplo

O exemplo a seguir demonstra como as informações nos arquivos contidos nos arquivos compactados, criados pela exportação do aplicativo de carga de trabalho a partir do Dynamic Workload Console, são usadas para ler os arquivos para implementação no ambiente de destino.

Tabela 63. Resolvendo o Arquivo de Mapeamento

Arquivo de definições

```
<model:JobStream carryforward=
  "true" draft="false"
 iskey="false" limit="55" name=
  "$JOBSTREAM_BADPBN34_JS1_1I$"
 onrequest="false" priority="100"
 workstation="$WORKSTATION_
  BADPBN34_WC1$">
     <model:runcycles/>
     <model:matching>
       <model:sameDay/>
     </model:matching>
     <model:restrictions/>
     <model:dependencies/>
     <model:jobs>
     <model:job confirmed="false"
definition=
   "$WORKSTATION_MDM112097$
  #$JOB_BADPBN34_J1$"
 isCritical="false" iskey="false"
  name="$JOB_BADPBN34_J1$"
priority="10">
       <model:restrictions/>
       <model:dependencies>
        <model:predecessor target=
         "$WORKSTATION_BADPBN34_
         WC1$#$JOBSTREAM_BADPBN34_
         JS2$.0">
          <model:matching>
            <model:previous>
          </model:matching>
       </model:predecessor>
      </model:dependencies>
    </model:job>
   </model:jobs>
</model:JobStream>
```

Arquivo de Mapeamento

```
#Nomes de estações de trabalho
#Substitua o valor pelo nome
 de uma estação de trabalho já
 existente no ambiente de destino.
#Consulte o arquivo MAIN_TEMPLATE_
  SourceEnv_reference.txt para
  obter detalhes sobre a
  estação de trabalho.
#
#Esta estação de trabalho é do tipo
  Gerenciador
WORKSTATION_MDM112097=MDM112097
#Esta estação de trabalho é do tipo
  Agente
WORKSTATION_MDM112097_1=
  MDM112097_1
#Esta estação de trabalho é do tipo
  Broker
WORKSTATION_MDM112097_DWB=
  MDM112097_DWB
```

Arquivo de informações de referência

```
CPUNAME $WORKSTATION_MDM112097$
  DESCRIPTION "Sample master domain
   manager"
  OS UNIX
  NODE MDM112097.romelab.it.ibm.com
   TCPADDR 35111
  TIMEZONE Europe/Rome
  DOMAIN MASTERDM
  FOR MAESTRO
    TYPE MDM
    AUTOLINK ON
    BEHINDFIREWALL ON
    FULLSTATUS ON
    SERVER A
END
```

Tabela 63. Resolvendo o Arquivo de Mapeamento (continuação)

Arquivo de definições	Arquivo de Mapeamento	Arquivo de informações de referência
O arquivo de informações de referência indica que a estação de trabalho denominada MDM112097 é do tipo gerenciador de domínio principal, em execução em um sistema operacional UNIX. Como o arquivo de definições contém referências ao nome da estação de trabalho, a entrada no arquivo de mapeamento `WORKSTATION_MDM112097=MDM112097` deverá ser atualizada. Substitua `MDM112097` pelo nome de uma estação de trabalho que já exista no ambiente de destino e que possua as mesmas características daquelas descritas no arquivo de informações de referência.		

Arquivo de definições

```
<model:JobStream carryforward="true" draft="false"
 iskey="false" limit="55" name="$JOBSTREAM_BADPBN34_JS1_1I$"
 onrequest="false" priority="100"
 workstation="$WORKSTATION_BADPBN34_WC1$">
     <model:runcycles/>
     <model:matching>
       <model:sameDay/>
     </model:matching>
     <model:restrictions/>
     <model:dependencies/>
     <model:jobs>
     <model:job confirmed="false"
defintion="$WORKSTATION_MDM112097$#$JOB_BADPBN34_J1$"
 isCritical="false" iskey="false" name="$JOB_BADPBN34_J1$"
priority="10">
       <model:restrictions/>
       <model:dependencies>
         <model:predecessor target="$WORKSTATION_BADPBN34_WC1$#$JOBSTREAM_
                                                          BADPBN34_JS2$.@">
          <model:matching>
           <model:previous>
          </model:matching>
       </model:predecessor>
      </model:dependencies>
     </model:job>
    </model:jobs>
</model:JobStream>
```

Implementando um aplicativo de carga de trabalho

A implementação de um aplicativo de carga de trabalho é um processo de duas etapas que começa com a customização do arquivo de mapeamento e, em seguida, a importação do arquivo de mapeamento e do arquivo de definições para o novo ambiente do Tivoli Workload Scheduler.

1. Extraia o conteúdo do modelo do aplicativo de carga de trabalho que contém o arquivo de definições, o arquivo de mapeamento e o arquivo de informações de referência do arquivo compactado criado pela operação de exportação do Dynamic Workload Console.
2. Customize o arquivo de mapeamento. Para cada objeto listado no arquivo de mapeamento, designe o nome de um objeto existente no ambiente de destino, ou renomeie-o com o nome que deseja que o objeto tenha no ambiente de destino. Consulte “Resolvendo o Arquivo de Mapeamento” na página 331 para obter informações sobre como customizar o arquivo de mapeamento.
3. Na linha de comandos, envie o seguinte comando, indicando os nomes de arquivos do arquivo de definições e do arquivo de mapeamento customizado:

   ```
   wappman -import <definition_xml_file> <mapping_properties_file>
   ```

   Consulte “Comando wappman” na página 335 para obter detalhes adicionais sobre o uso e sintaxe do comando.

Todos os objetos do Tivoli Workload Scheduler definidos no *definition_xml_file* serão criados no ambiente de destino se ainda não existirem. Atualize o arquivo de

mapeamento para resolver as referências aos objetos que precisam ser customizados para refletir o ambiente de destino no qual o aplicativo de carga de trabalho será implementado.

Subsequentemente, é possível atualizar um aplicativo de carga de trabalho. Existem duas maneiras nas quais oaplicativo de carga de trabalho pode ser atualizado ou modificado:

Modificando o modelo no ambiente de origem
Uma versão atualizada do modelo pode ser implementada novamente no ambiente de destino. Os objetos já presentes no banco de dados do Tivoli Workload Scheduler do ambiente de destino são substituídos pelas versões atualizadas, os objetos que ainda não existem no ambiente de destino são criados, e os objetos serão excluídos do ambiente de destino se a definição de objeto tiver sido removida do aplicativo de carga de trabalho atualizado. O mesmo arquivo de mapeamento usado para implementar originalmente oaplicativo de carga de trabalho pode ser usado para atualizá-lo, customizando novos objetos que estão sendo implementados com a atualização.

Modificando a instância no ambiente de destino
Após a implementação de um aplicativo de carga de trabalho, é possível incluir uma nova tarefa em um fluxo de tarefas, modificar uma definição de tarefa ou remover uma tarefa ou fluxo de tarefas. No entanto, essas mudanças não serão mantidas se oaplicativo de carga de trabalho for atualizado com o modelo revisado do aplicativo de carga de trabalho.

Comando wappman

Cria, substitui, exclui, exibe e lista um aplicativo de carga de trabalho.

Nota: No Windows 2012, o comando não é suportado no Windows PowerShell.

Autorização

Você deve ter acesso ***add*** se criar um novoaplicativo de carga de trabalho. Se o objeto já existir no banco de dados, você deve ter o acesso ***modify*** para o objeto.

Sintaxe

wappman[*connection_parameters*]
[-u]|[-V]
| **{-import|-replace}** <*definition_xml_file*> <*mapping_properties_file*
-list
-delete <*workload_application_name*>
-display <*workload_application_name*>
]

Argumentos

[*connection_parameters***]**
Um arquivo com os parâmetros de conexão a ser usado para conectar-se ao servidor. É possível usar parâmetros de conexão para substituir os valores especificados nos arquivos useropts e localopts. Para determinar quais propriedades da conexão usar, é feita uma verificação na seguinte ordem:

1. Parâmetros especificados na própria sequência de caracteres de comando.

2. Parâmetros especificados no arquivo de propriedades customizado.
3. O arquivo useropts.
4. O arquivo localopts.

Os valores válidos incluem:

[-file <*custom_properties_file***>]**

[-host <*nome do host***>]**

[-port <*port_number***>]**

[-protocol {http | https}]

[-password <*password***>]**

[-username <*user_name***>]**
em que *user_name* é um usuário do Tivoli Workload Scheduler com privilégios suficientes para executar a operação.

-u Exibe informações de utilização do comando e sai.

-V Exibe a versão do comando e sai.

{-import|-replace} <*definition_xml_file***> <***mapping_properties_file***>**
O nome do arquivo de definições e do arquivo de mapeamento a serem usados durante a importação ou substituição de um aplicativo de carga de trabalho. A operação de importação de um aplicativo de carga de trabalho é igual à implementação de um aplicativo de carga de trabalho.

import
Depois de customizar o arquivo de mapeamento, o arquivo de mapeamento, juntamente com o arquivo de definições são transmitidos como entrada para o comando **wappman -import** para implementar oaplicativo de carga de trabalho em um ambiente Tivoli Workload Scheduler diferente do ambiente de origem no qual o modelo do aplicativo de carga de trabalho foi originalmente criado.

replace
Uma versão atualizada do modelo do aplicativo de carga de trabalho é reimportada para o ambiente de destino. Os objetos que foram inicialmente criados no banco de dados do Tivoli Workload Scheduler quando o modelo do aplicativo de carga de trabalho foi implementado pela primeira vez serão atualizados se ainda estiverem presentes no banco de dados, criados se não estiverem mais presentes e excluídos se não estiverem mais presentes no modelo do aplicativo de carga de trabalho atualizado.

A operação de substituição falhará se um objeto externo aoaplicativo de carga de trabalho referenciar um objeto no aplicativo de carga de trabalho e, com a operação de substituição, o objeto referenciado será excluído, porque não está mais presente no aplicativo de carga de trabalho atualizado.

Se foram feitas referências do aplicativo de carga de trabalho para um elemento externo, a referência será excluída com a ação substituir.

O mesmo arquivo de mapeamento usado para implementar originalmente oaplicativo de carga de trabalho pode ser usado para atualizá-lo, fazendo as mudanças necessárias para refletir oaplicativo de carga de trabalho atualizado. O arquivo de

mapeamento juntamente com o arquivo de definições são transmitidos como entrada para o comando **wappman -replace**.

-list Lista todos os aplicativos de carga de trabalho presentes no ambiente.

-delete <*workload_application_name***>**
Exclui oaplicativo de carga de trabalho especificado e todos os objetos que foram incluídos no ambiente quando oaplicativo de carga de trabalho foi originalmente implementado.

-display <*workload_application_name***>**
Exibe o conteúdo do aplicativo de carga de trabalho especificado.

Exibe todos os objetos contidos no aplicativo de carga de trabalho que foram criados durante o processo de importação.

Logs e Rastreios

É possível configurar a linha de comandos **wappman**, modificando os parâmetros no arquivo `CLI.ini` localizado no caminho*TWA_home*`/TWS/ITA/cpa/config`. Este arquivo contém parâmetros para os arquivos de logs e de rastreio de mensagens relacionados aoaplicativos de carga de trabalho. Você deve alterar os parâmetros apenas se for recomendado que faça isso na documentação do Tivoli Workload Scheduler ou se for solicitado pelo Suporte de Software IBM.

Exemplos

Para importar um aplicativo de carga de trabalho para um novo ambiente, execute:

```
wappman -import <definition_xml_file> <mapping_properties_file>
```

Para substituir um aplicativo de carga de trabalho existente, execute:

```
wappman -replace <definition_xml_file> <mapping_properties_file>
```

Para excluir um aplicativo de carga de trabalho, execute:

```
wappman -delete <workload_application_name>
```

Para listar todos os aplicativos de carga de trabalho disponíveis em um ambiente, execute:

```
wappman -list
```

Para exibir um aplicativo de carga de trabalho específico, execute:

```
wappman -display <workload_application_name>
```

Consulte também

Para obter informações sobre como definir um modelo do aplicativo de carga de trabalho no ambiente de origem, consulte “Criando um Modelo do aplicativo de carga de trabalho” na página 272.

Consulte “Implementando um aplicativo de carga de trabalho” na página 334 para obter informações adicionais sobre como gerenciar um aplicativo de carga de trabalho no ambiente de destino.

Capítulo 11. Gerenciando Objetos no Plano - conman

O ambiente do plano de produção do Tivoli Workload Scheduler é gerenciado utilizando o programa de linha de comandos **conman**. O programa **conman** é utilizado para iniciar e parar o processamento, alterar e exibir o plano de produção Symphony e controlar a vinculação da estação de trabalho em uma rede. Pode ser utilizado a partir do gerenciador de domínio principal e de qualquer agente tolerante a falhas na rede do Tivoli Workload Scheduler. Este capítulo está dividido nas seções a seguir:

- "Configurando o Programa de linha de Comandos conman"
- "Executando comandos a partir do conman" na página 343
- "Selecionando Tarefas em Comandos" na página 345
- "Selecionando Fluxos de Tarefas em Comandos" na página 353
- "Gerenciando Tarefas e Fluxos de Tarefas a partir de Agentes de Nível Anterior" na página 360
- "Conman commands" na página 360

Configurando o Programa de linha de Comandos conman

O programa de linha de comandos **conman** gerencia o ambiente do plano de produção.

É possível utilizar o programa **conman** a partir do gerenciador de domínio principal e a partir de qualquer estação de trabalho do agente tolerante a falhas na rede do Tivoli Workload Scheduler.

Configurando o Ambiente do conman

Esta seção fornece informações sobre a configuração que pode ser escolhida para o ambiente **conman**.

Nota: Nos sistemas Windows, antes de executar **conman**, assegure que a página de códigos e as fontes estejam configuradas corretamente no shell DOS para evitar conversão de caracteres inválida. Para obter informações adicionais sobre as configurações necessárias, consulte a seção *Internationalization notes* no *IBM Tivoli Workload Scheduler: Liberação Notes* em http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27041032.

Saída do Terminal

A saída para seu computador é determinada pelas variáveis shell chamadas ***MAESTROLINES*** e ***MAESTROCOLUMNS***. Se ainda não estiverem definidas, as variáveis shell padrão, ***LINES*** e ***COLUMNS***, serão utilizadas. As variáveis podem ser definidas como a seguir:

MAESTROLINES
: Especifica o número de linhas por tela. O padrão é **24**. No final de cada pagina de tela, o **conman** solicita para continuar. Se ***MAESTROLINES*** (ou ***LINES***) estiver definido como zero ou um número negativo, o **conman** não será pausado no final de uma página.

© Copyright IBM Corp. 1999, 2014

O uso de MAESTROLINES é recomendado, pois a variável LINES é uma variável de sistema operacional shell e na maioria dos sistemas operacionais ela é automaticamente reconfigurada pelo próprio sistema operacional.

MAESTROCOLUMNS
Especifica o número de caracteres por linha. As seguintes opções estão disponíveis:

- Menos que 120
- Igual ou mais que 120

MAESTRO_OUTPUT_STYLE
Especifica como será feita a exibição dos nomes de objetos. Se definidos como **LONG**, serão exibidos nomes completos. Se configurados para qualquer valor que não seja **LONG**, nomes longos serão truncados para oito caracteres seguidos por um sinal de mais (+).

Saída Off-line

A opção **;off-line** nos comandos do **conman** é utilizada, geralmente, para imprimir a saída de um comando. Ao incluí-la, as seguintes variáveis shell controlam a saída:

MAESTROLP
Especifica o destino da saída de um comando. Defina-o como um dos seguintes:

> *file* Redireciona a saída para um arquivo e sobrescreve seu conteúdo. Se o arquivo não existir, ele será criado.

>> *file* Redireciona a saída para um arquivo, anexando-a ao final desse arquivo. Se o arquivo não existir, ele será criado.

| *command*
Canaliza a saída para um comando do sistema ou processo. O comando do sistema é executado independentemente da geração da saída.

|| *command*
Canaliza a saída para um comando do sistema ou processo. O comando do sistema não será executado, se não houver saída.

O valor padrão para ***MAESTROLP*** é **| lp -tCONLIST**, o que canaliza a saída do comando para a impressora e coloca o título "CONLIST" na página do banner da impressão.

MAESTROLPLINES
Especifica o número de linhas por página. O padrão é **60**.

MAESTROLPCOLUMNS
Especifica o número de caracteres por linha. O padrão é **132**.

As variáveis devem ser exportadas antes da execução do **conman**.

Selecionando o Prompt do conman no UNIX

O prompt de comandos do **conman** é, por padrão, um sinal de porcentagem (%). Isso está definido no arquivo *TWS_home*/localopts. O prompt de comandos padrão é um traço (-). Para selecionar um prompt diferente, edite a opção **conman** prompt no arquivo localopts e altere o traço. O prompt pode ter até 10 caracteres de comprimento, sem incluir o sinal de sustenido (#) final requerido.

```
#--------------------------------------------------------------------------
# Atributos de formato customizado
#
```

```
date format =    1   # The possible values are 0-ymd, 1-mdy,
2-dmy, 3-NLS.
composer prompt =   -
conman prompt =    %
switch sym prompt =  <n>%
#-------------------------------------------------------------------------------
```

Executando conman

Para configurar o ambiente para utilizar **conman**, configure as variáveis *PATH* e *TWS_TISDIR* executando um dos seguintes scripts:

No UNIX:

- **. ./***TWS_home*/**tws_env.sh** para shells Bourne e Korn
- **. ./***TWS_home*/**tws_env.csh** para shells C

No Windows:

- *TWS_home***tws_env.cmd**

Em seguida, utilize a sintaxe a seguir para executar comandos a partir da interface com o usuário **conman**:

conman [*connection_parameters*] [**"***command*[**&**[*command*]...] [**&**]**"**]

em que:

connection_parameters

Se estiver usando**conman** a partir do gerenciador de domínio principal, os parâmetros de conexão foram configurados na instalação e não precisam ser fornecidos, a menos que você não deseje usar os valores padrão.

Se estiver usando**conman** a partir do cliente da linha de comandos em outra estação de trabalho, os parâmetros de conexão podem ser fornecidos por um ou mais destes métodos:

- Armazenados no arquivo `localopts`
- Armazenados no arquivo `useropts`
- Fornecidos ao comando em um arquivo de parâmetro
- Fornecidos ao comando como parte da sequência de caracteres de comando

Para obter uma visão geral dessas opções, consulte “Configurando Opções para Utilização das Interfaces com o Usuário” na página 53. Para obter detalhes integrais dos parâmetros de configuração, consulte o tópico sobre como configurar o acesso do cliente de linha de comandos no *Tivoli Workload Scheduler: Administration Guide*.

É possível chamar a linha de comandos **conman** no modo*em lote* e *interativo*.

Ao executar **conman** no modo*interativo*, você, primeiramente, ativa o programa de linha de comandos **conman** e, em seguida, a partir do prompt da linha de comandos **conman**, executa comandos um por vez, por exemplo:

```
conman –username admin2 –password admin2pwd
   ss @+state=hold;deps
   dds sked5;needs=2 tapes
```

Ao executar **conman** no modo*em lote*, você primeiro ativa o programa de linha de comandos **conman** especificando como parâmetro de entrada o comando a ser emitido. Depois de o comando ser processado, o programa de linha de comandos **conman** sai, por exemplo

```
conman"sj&sp"
```

Ao emitir comandos a partir de **conman** no modo em lote, certifique-se de colocar os comandos entre aspas duplas. Seguem exemplos de como utilizar o modo de batch para emitir mais de um comando a partir de **conman**:

- O **conman** executa os comandos **sj** e **sp** e, em seguida, sai:

  ```
  conman"sj&sp"
  ```

- O **conman** executa os comandos **sj** e **sp** e, em seguida, solicita um comando:

  ```
  conman "sj&sp&"
  ```

- O **conman** lê os comandos a partir do arquivo **cfile**:

  ```
  conman < cfile
  ```

- Os comandos do arquivo **cfile** são canalizados para o **conman**:

  ```
  cat cfile | conman
  ```

Nota: Nas estações de trabalho Windows, se o UAC (User Account Control) estiver ligado e a lista de exceções do UAC não contiver o arquivo `cmd.exe`, você deve abrir a shell do prompt de comandos do DOS com a opção `"Executar como Administrador"` para executar o**conman** em sua estação de trabalho como um usuário genérico diferente do Administrador ou o usuário do Tivoli Workload Scheduler.

Caracteres de Controle

É possível digitar os seguintes caracteres de controle para interromper o **conman**.

Control+c
: O **conman** pára a execução do comando atual na próxima etapa que puder ser interrompida e retorna um prompt de comandos.

Control+d
: **conman** é encerrado após executar o comando atual somente em estações de trabalho UNIX.

Executando Comandos do Sistema

Ao inserir um comando do sistema utilizando um canal ou um prefixo de comando do sistema (: ou !), ele é executado por um processo-filho. O ID do usuário efetivo do processo-filho é configurado para o ID do usuário que está executando**conman** para evitar as violações de segurança.

Prompt do Usuário

Quando você utiliza caracteres curinga para selecionar os objetos sobre os quais a ação será executada por um comando, o **conman** solicita confirmação depois de localizar cada objeto correspondente. Responder **sim** permite que a ação seja tomada e **não** pula o objeto sem qualquer ação.

Ao executar **conman** de forma interativa, os prompts de confirmação são emitidos em seu computador. Pressionar a tecla **Return** em resposta a um prompt é interpretado como **não**. As solicitações através de prompt podem ser desativadas, incluindo-se a opção **;noask** em um comando.

Embora nenhum aviso de confirmação seja emitido quando o**conman** não está sendo executado no modo interativo, ele age como se a resposta tivesse sido **não** em cada caso, e nenhum objeto recebe ação. É importante, portanto, incluir a opção **;noask** nos comandos quando o**conman** não for executado no modo interativo.

Executando comandos a partir do conman

Os comandos do **conman** consistem nos seguintes elementos:

commandname selection arguments

em que:

commandname
Especifica o nome do comando.

selection
Especifica o objeto ou conjunto de objetos sobre os quais a ação será tomada.

argumentos
Especifica os argumentos do comando.

A seguir, um exemplo de comando do **conman**:

```
sj sked1(1100 03/05/2006).@+state=hold~priority=0;info;offline
```

em que:

sj A forma abreviada do comando **showjobs**.

sked1(1100 03/05/2006).@+state=hold~priority=0
Seleciona todas as tarefas do fluxo de tarefas **sked1(1100 03/05/2006)** que estão no estado HOLD com uma prioridade diferente de zero.

;info;off-line
Argumentos do comando **showjobs**.

Curingas

Os seguintes caracteres curinga são permitidos:

@ Substitui um ou mais caracteres alfanuméricos.

? Substitui um caractere alfanumérico.

% Substitui um caractere numérico.

Delimitadores e caracteres especiais

A Tabela 64 lista os caracteres que têm significados especiais em comandos **conman**:

Tabela 64. Delimitadores e Caracteres Especiais para conman

Caractere	Descrição
&	Delimitador de comando. Consulte "Configurando o Programa de linha de Comandos conman" na página 339.
+	Um delimitador utilizado para selecionar objetos para comandos. Ele adiciona um atributo que o objeto deve ter. Por exemplo: `sked1(1100 03/05/2006).@~priority=0`
~	Um delimitador utilizado para selecionar objetos para comandos. Ele adiciona um atributo que o objeto não deve ter. Por exemplo: `sked1(1100 03/05/2006).@~priority=0`
;	Delimitador de argumento. Por exemplo: `;info;off-line`
,	Delimitador de repetição e de faixa. Por exemplo: `state=hold,sked,pend`
=	Delimitador de valor. Por exemplo: `state=hold`

Tabela 64. Delimitadores e Caracteres Especiais para conman (continuação)

Caractere	Descrição
:!	Prefixos de comandos que transmitem o comando para o sistema. Esses prefixos são opcionais; se o **conman** não reconhecer o comando, ele será transmitido automaticamente para o sistema. Por exemplo: `!ls ou :ls`

*	Prefixo do comentário. O prefixo deve ser o primeiro caractere em uma linha de comandos ou após um delimitador de comandos. Por exemplo:
	`*comment`
	
	ou
	`sj& *comment`

>	Redireciona a saída de comando para um arquivo e sobrescreve seu conteúdo. Se o arquivo não existir, ele será criado. Por exemplo: `sj> joblist`
>>	Redireciona a saída de comando para um arquivo, anexando-a ao final desse arquivo. Se o arquivo não existir, ele será criado. Por exemplo: `sj >> joblist`
|	Canaliza a saída de comando para um comando do sistema ou processo. O comando do sistema é executado independentemente da geração da saída. Por exemplo: `sj| grep ABEND`
||	Canaliza a saída de comando para um comando do sistema ou processo. O comando do sistema não será executado, se não houver saída. Por exemplo: `sj || grep ABEND`

Processamento de comandos do conman

O programa **conman** executa os comandos que alteram os status dos objetos, como start ou stop para uma estação de trabalho, e os comandos que modificam objetos no plano de maneira *assíncrona*. Isso significa que será possível observar um retardo entre o tempo de envio do comando e o tempo que as informações armazenadas no arquivo `Symphony` são atualizadas com o resultado do comando.

Isso ocorre porque o programa **conman** não atualiza as informações armazenadas no arquivo `Symphony`; o **conman** submete os comandos ao **batchman**, que é o único processo que pode acessar e atualizar as informações contidas no arquivo `Symphony`. Por isso, é necessário aguardar que o **batchman** processe o pedido de modificação do objeto emitido pelo **conman** e, em seguida, atualizar as informações sobre o objeto armazenado no arquivo `Symphony` antes de ver as informações atualizadas na saída do comando ***showobj***.

As mudanças feitas usando o programa conman que afetam o arquivo `Symphony` também são aplicadas às informações do plano replicado no banco de dados.

Por exemplo, se você pedir para excluir uma dependência utilizando o comando **conman deldep**, **conman** envia o comando **deldep** postando um evento na caixa de correio `Mailman.msg`. O processo **mailman** obtém as informações sobre o pedido de exclusão de `Mailman.msg` e as coloca na caixa de correio `Intercom.msg` na estação de trabalho que possui o recurso do qual você exclui a dependência. Em cada estação de trabalho, o **batchman** recebe os eventos em sua caixa postal `Intercom.msg` e os processa na mesma ordem em que foram recebidos. Se **batchman** estiver ocupado por qualquer motivo, os eventos que transportam pedidos para executar os comandos **conman** continuam a ser enfileirados no arquivo `Intercom.msg` aguardando para serem lidos e processados pelo **batchman**.

Além disso, quando o **batchman** processa o evento, o operador não é notificado. Como resultado, uma dependência poderia ter sido excluída, embora parecesse que não, porque o **batchman** estava muito ocupado para executar a operação solicitada imediatamente. Se você executar o comando novamente, a exclusão já poderá ter sido bem-sucedida, apesar da exibição de uma mensagem informando que o comando foi encaminhado com êxito para o **batchman** no prompt do **conman**.

Selecionando Tarefas em Comandos

Para comandos que operam em tarefas, os jobs de destino selecionados por intermédio de atributos e qualificadores. A sintaxe de seleção do job é mostrada abaixo e descrita nas páginas a seguir.

Sintaxe

[*workstation***#**]{*jobstreamname*(*hhmm*[*date*]).*jobname* | *jobnumber*} [{**+** | **~**}*jobqualifier*[...]]

ou

[*workstation***#**]*jobstream_id.job* [{**+** | **~**]*jobqualifier*[...]]**;schedid**

ou:

netagent::[*workstation***#**]{*jobstreamname*(*hhmm*[*date*]).*jobname* | *jobstream_id.jobname***;schedid**}

Argumentos

estação de trabalho
Quando utilizado com *jobstream.job*, especifica o nome da estação de trabalho na qual o fluxo de tarefa é executado. Quando utilizado com *jobnumber*, especifica a estação de trabalho na qual o job é executado. Exceto quando também utilizar `schedid`, caracteres curingas são permitidos. Esse argumento pode ser necessário dependendo da estação de trabalho onde o comando é ativado, da seguinte forma:

- Se você ativar o comando na estação de trabalho onde as tarefas de destino foram executadas, o argumento *workstation* é opcional.
- Se você ativar o comando em uma estação de trabalho hospedada, o argumento *workstation* é obrigatório. Estações de trabalho hospedadas são:
  - agentes estendidos
  - agentes
  - conjuntos
  - conjuntos dinâmicos

jobstreamname
Especifica o nome do fluxo de tarefas no qual a tarefa é executada. Caracteres curinga são permitidos.

(*hhmm* **[***date***])**
Indica a hora e a data que a instância do fluxo de tarefas é localizada no plano de pré-produção. O valor *hhmm* corresponde ao valor designado à palavra-chave **schedtime** na definição do fluxo de tarefas se nenhuma restrição de tempo **at** tiver sido configurada. Após a instância do fluxo de tarefas ter iniciado o processamento, o valor de ***hhmm* [*date*]** é configurado para a hora de início do fluxo de tarefas. A utilização de curingas nesse campo não é permitida. Ao emitir comandos **conman** seqüenciais a partir do prompt de shell, coloque o comando **conman** entre aspas duplas **"**. Por exemplo, execute esse comando da seguinte forma:

```
conman "sj my_workstation#my_js(2101 02/23).@"
```

jobstream_id
Especifica o identificador do fluxo de tarefa exclusivo. Consulte "Argumentos" na página 354 para obter informações adicionais sobre identificadores de fluxo de tarefas.

schedid Indica que o identificador do fluxo de tarefa é utilizado na seleção do fluxo de tarefa.

jobname
Especifica o nome do job. Caracteres curinga são permitidos.

jobnumber
Especifica o número do job.

jobqualifier
Consulte a seção a seguir .

netagent
Especifica o nome do agente de rede do Tivoli Workload Scheduler que faz a interface com a rede remota do Tivoli Workload Scheduler que contém a tarefa de destino. Os dois-pontos (::) são requeridos como delimitador. Caracteres curinga são permitidos. Para obter informações adicionais, consulte Capítulo 18, "Gerenciando Dependências da Interligação de Redes", na página 615.

Nota: O Tivoli Workload Scheduler ajuda a identificar a instância correta do fluxo de tarefas quando a seleção de fluxo de tarefas fornece um resultado ambíguo se mais de uma instância atender seus critérios de seleção. Por exemplo, quando mais de uma instância de `WK1#J1` é incluída no plano de produção e, portanto, a seleção do fluxo de tarefas fornece um resultado ambíguo, o prompt a seguir é gerado automaticamente para permitir que você escolha a instância correta:

```
Process WK1#J1[(0600 03/04/06),(0AAAAAAAAAAAABTB)]
        (enter "y" for yes, "n" for no)?y
Command forwarded to batchman for WK1#J1[(0600 03/04/06),(0AAAAAAAAAAAABTB)]
Process WK1#J1[(1010 03/04/06),(0AAAAAAAAAAAABTC)]
        (enter "y" for yes, "n" for no)?n
```

Na saída, somente a instância do fluxo de tarefas planejada em `(0600 03/04/06)` e com o identificador `0AAAAAAAAAAAABTB` é selecionada para executar o comando.

Qualificadores de Tarefa

Os qualificadores de job especificam os atributos de jobs nos quais os comandos atuarão. Eles podem ter como prefixo + ou ~. Se um qualificador de job for precedido por +, os jobs contendo esse atributo específico serão selecionados para execução do comando. Se um qualificador de job for precedido por ~, os jobs contendo esse atributo específico serão excluídos da execução do comando.

As palavras-chave do qualificador de job podem ser abreviadas com o mínimo de caracteres iniciais, o necessário para distingui-las exclusivamente umas das outras.

at[=*time* | *lowtime,* | *,hightime* | *lowtime*,*hightime* **]**
Seleciona ou exclui tarefas com base na hora especificada na dependência **at**.

time

Especifica a hora da seguinte forma:

hhmm[**+***n* **days** | *date*] [**timezone**|**tz** *tzname*]

em que:

hhmm A hora e minuto.

+*n* **days**
A próxima ocorrência de *hhmm* em *n* número de dias.

date A próxima ocorrência de *hhmm* em *date*, expressa como *mm/dd[/yy]*.

timezone | tz *tzname*
O nome do fuso horário do job. Consulte Capítulo 16, "Gerenciando fusos horários", na página 597 para obter nomes válidos.

time obedece as seguintes regras:

- Quando *hhmm* é anterior à hora atual, a hora de início é amanhã; quando *hhmm* é posterior à hora atual, a hora de início é hoje.
- Quando *hhmm* é maior que 2400, ele é dividido por 2400. Do resultado, a parte total representa o número de + *dias* e a parte decimal representa a hora.

lowtime
Especifica o limite inferior de um intervalo de tempo, expresso no mesmo formato de *time*. São selecionadas tarefas que estão planejadas para iniciar após essa hora.

hightime
Especifica o limite superior de um intervalo de tempo, expresso no mesmo formato de *time*. São selecionadas tarefas que estão planejadas para iniciar antes dessa hora.

Se **at** for utilizado sozinho e for precedido por +, então, as tarefas selecionadas são aquelas que contêm uma dependência **at**.

Se **at** for utilizado sozinho e for precedido por ~, então, as tarefas selecionadas são aquelas que não contém uma dependência **at**.

confirmed
Seleciona ou exclui tarefas que foram planejadas utilizando-se a palavra-chave **confirm**.

critical
Seleciona ou exclui tarefas que foram sinalizadas com a palavra-chave **critical** em uma definição de fluxo de tarefas.

critnet Seleciona ou exclui tarefas que são sinalizadas como **critical** ou são predecessoras de tarefas críticas. Portanto, aplica-se a todas as tarefas que têm um horário de início crítico configurado.

O horário de início crítico de uma tarefa crítica é calculado subtraindo sua duração estimada do respectivo prazo final. O horário de início crítico de um predecessor é calculado subtraindo sua duração estimada do horário de início crítico de seu sucessor. Em uma rede crítica, os horários de início críticos são calculados começando pela tarefa crítica e trabalhando de modo retroativo ao longo da linha dos respectivos predecessores.

deadline[=*time* **|** *lowtime***, |** *,hightime* **|** *lowtime,hightime***]**
Especifica o tempo no qual um job deve ser concluído.

hhmm[**+***n* **days** | *date*] [**timezone | tz** *tzname*]

hhmm A hora e minuto.

+*n* **days**
Um deslocamento em dias da hora do prazo final planejada.

date A próxima ocorrência de *hhmm* em *date*, expressa como *mm/dd[/yy]*.

timezone | tz *tzname*
Especifica o fuso horário a ser utilizado ao calcular o prazo final. Consulte Capítulo 16, "Gerenciando fusos horários", na página 597 para obter nomes de fuso horários. O padrão é o fuso horário da estação de trabalho em que a tarefa ou fluxo de tarefas é lançado.

lowtime
Especifica o limite inferior de um intervalo de tempo, expresso no mesmo formato de *time*. As tarefas selecionadas têm um prazo final planejado não anterior a essa hora.

hightime
Especifica o limite superior de um intervalo de tempo, expresso no mesmo formato de *time*. As tarefas selecionadas têm um prazo final planejado não posterior a essa hora.

every[=*rate* | *lowrate*, | ,*highrate* | *lowrate*,*highrate***]**
Seleciona ou exclui jobs se tiverem ou não uma taxa de repetição.

rate Especifica a taxa de execução planejada, expressa como horas e minutos (*hhmm*).

lowrate Especifica o limite inferior de uma faixa de taxas, expressa no mesmo formato de *rate*. São selecionados jobs que têm taxas de repetição iguais ou maiores que essa taxa.

highrate
Especifica o limite superior de uma faixa de taxas, expressa no mesmo formato de *rate*. São selecionados jobs que têm taxas de repetição menores ou iguais a essa taxa.

Se **every** for utilizado sozinho e estiver precedido por +, os jobs selecionados serão aqueles que contêm qualquer taxa de repetição.

Se **every** for utilizado sozinho e estiver precedido por ~, os jobs selecionados serão aqueles que não contêm nenhuma taxa de repetição.

finished[=*time* | *lowtime*, | ,*hightime* | *lowtime*,*hightime***]**
Seleciona ou exclui jobs se tiverem terminado ou não.

time Especifica a hora exata em que o job terminou, expressa como a seguir:

hhmm [*date*] [**timezone | tz** *tzname*]

hhmm A hora e minuto.

date A próxima ocorrência de *hhmm* na data, expressa como *mm/dd[/yy]*.

timezone | tz *tzname*
O nome do fuso horário do job. Consulte Capítulo 16, "Gerenciando fusos horários", na página 597 para obter nomes válidos.

lowtime
Especifica o limite inferior de um intervalo de tempo, expresso no mesmo formato de *time*. São selecionados jobs que terminaram nesta hora ou após ela.

hightime
Especifica o limite superior de um intervalo de tempo, expresso no mesmo formato de *time*. São selecionados jobs que terminaram nesta hora ou antes dela.

Se **finished** for utilizado sozinho e estiver precedido por +, os jobs selecionados serão aqueles cuja execução terminou.

Se **finished** for utilizado sozinho e estiver precedido por ~, os jobs selecionados serão aqueles cuja execução não terminou.

follows=[*netagent***::][***workstation***#]{***jobstreamname***(***hhmm* **[***mm/dd***[/***yy***]])[.***job* | **@] |** *jobstream_id***.***job***;schedid}|** *job***[;nocheck][,...]**
Seleciona ou exclui as tarefas se tiverem ou não uma dependência `follows`.

netagent
Especifica o nome do agente de rede do Tivoli Workload Scheduler que faz a interface com a rede remota do Tivoli Workload Scheduler que contém a tarefa pré-requisita. Caracteres curinga são permitidos. Para obter informações adicionais, consulte Capítulo 18, “Gerenciando Dependências da Interligação de Redes”, na página 615.

estação de trabalho
Especifica o nome da estação de trabalho na qual o job de pré-requisito é executado. Caracteres curinga são permitidos.

jobstreamname
Especifica o nome do fluxo de tarefas no qual a tarefa de pré-requisito é executada. Caracteres curinga são permitidos. Se você digitar *jobstreamname***.@**, você especifica que a tarefa de destino segue todas as tarefas do fluxo de tarefa.

jobname
Especifica o nome do job de pré-requisito. Quando digitado sem um *jobstreamname*, significa que a tarefa de pré-requisito está no mesmo fluxo de tarefa que a tarefa de destino. Não especifique o nome do job utilizando caracteres curinga para uma dependência follows.

jobstream_id
Especifica o identificador do fluxo de tarefa exclusivo. Consulte “Argumentos” na página 354 para obter informações adicionais sobre identificadores de fluxo de tarefas.

schedid
Essa palavra-chave, se presente, aplica-se a todos os identificadores de fluxos de tarefas especificados na cláusula e indica que, para todos os fluxos de tarefas especificados, você está utilizando os *jobstream_id*s, não os *jobstreamname*s. Se quiser selecionar alguns fluxos de tarefas utilizando o *jobstream_id* e alguns fluxos de tarefas utilizando o *jobstreamname*, você deverá utilizar duas cláusulas **follows** diferentes, uma contendo os fluxos de tarefas identificados pelas palavras-chave *jobstreamname* sem as palavras-chave **schedid** e a outra contendo os fluxos de tarefas identificados pelo *jobstream_id* com a palavra-chave **schedid**.

nocheck
É válido somente para comandos de envio e utilizado em conjunto com a palavra-chave **schedid**. A palavra-chave **nocheck** indica que **conman** não precisa verificar a existência da tarefa pré-requisito

jobstream_id.job no arquivo `Symphony`. Supõe-se que *jobstream_id.job* exista. Caso não exista, **batchman** imprimirá uma mensagem de aviso em `stdlist`.

Se **follows** for utilizado sozinho e estiver precedido por +, os jobs serão selecionados se contiverem dependências `follows`.

Se **follows** for utilizado sozinho e estiver precedido por ~, os jobs serão selecionados se não contiverem dependências `follows`.

logon=*username*
Seleciona jobs com base nos nomes de usuário sob os quais são executados. Se *username* contiver caracteres especiais, ele deverá ser colocado entre aspas ("). Caracteres curinga são permitidos.

needs[=[*workstation***#]***resourcename***]**
Seleciona ou exclui jobs se tiverem ou não uma dependência de recurso.

estação de trabalho
Especifica o nome da estação de trabalho na qual o recurso foi definido. Caracteres curinga são permitidos.

resourcename
Especifica o nome do recurso. Caracteres curinga são permitidos.

Se **needs** for utilizado sozinho e estiver precedido por +, os jobs serão selecionados se contiverem dependências `resource`.

Se **needs** for utilizado sozinho e estiver precedido por ~, os jobs serão selecionados se não contiverem dependências `resource`.

opens[=[*workstation***#]***filename***[(***qualifier***)]]**
Seleciona jobs se tiverem ou não uma dependência `file`. Uma dependência `file` ocorre quando um job ou fluxo de job depende da existência de um ou mais arquivos para poder começar a execução.

estação de trabalho
Especifica o nome da estação de trabalho na qual o arquivo existe. Caracteres curinga são permitidos.

nome do arquivo
Especifica o nome do arquivo. O nome deverá ser colocado entre aspas (") se contiver caracteres especiais diferentes do seguinte: alfanuméricos, traços (-), barras (/), barras invertidas (\) e sublinhados (_). Caracteres curinga são permitidos.

qualifier
Uma condição de teste válida. Se omitido, os jobs serão selecionados ou excluídos independente de um qualificador.

Se **opens** for utilizado sozinho e estiver precedido por +, os jobs serão selecionados se contiverem dependências `file`.

Se **opens** for utilizado sozinho e estiver precedido por ~, os jobs serão selecionados se não contiverem dependências `file`.

priority=*pri* | *lowpri***,** | **,***highpri* | *lowpri***,***highpri*
Seleciona ou exclui jobs com base em suas prioridades.

pri Especifica o valor de prioridade. Você pode digitar de **0** a **99**, **hi** ou **go**.

lowpri Especifica o limite inferior de uma faixa de prioridades. Os jobs são selecionados com prioridades iguais ou maiores que esse valor.

highpri Especifica o limite superior de uma faixa de prioridades. Os jobs são selecionados com prioridades menores ou iguais a esse valor.

prompt[=*promptname* | *msgnum***]**
Seleciona ou exclui jobs se tiverem ou não uma dependência `prompt`.

promptname
Especifica o nome de um prompt global. Caracteres curinga são permitidos.

msgnum
Especifica o número de mensagem de um prompt local.

Se **prompt** for utilizado sozinho e estiver precedido por +, os jobs serão selecionados se contiverem dependências `prompt`.

Se **prompt** for utilizado sozinho e estiver precedido por ~, os jobs serão selecionados se não contiverem dependências `prompt`.

recovery=*recv-option*
Seleciona ou exclui jobs com base em suas opções de recuperação.

recv-option
Especifica a opção de recuperação do job como **parar**, **continuar** ou **reexecutar**.

scriptname=*filename*
Seleciona ou exclui jobs com base nos nomes de seus arquivos executáveis.

nome do arquivo
Especifica o nome de um arquivo executável. O nome deverá ser colocado entre aspas (") se contiver caracteres especiais diferentes do seguinte: alfanuméricos, traços (-), barras (/), barras invertidas (\) e sublinhados (_). Caracteres curinga são permitidos.

started[=*time* | *lowtime***,** | **,***hightime* | *lowtime***,***hightime***]**
Seleciona ou exclui jobs se tiverem ou não sido iniciados.

time Especifica a hora exata em que o job foi iniciado, expressa como a seguir:

hhmm [*date*] [**timezone**|**tz** *tzname*]

hhmm A hora e minuto.

date A próxima ocorrência de *hhmm* na data, expressa como *mm/dd[/aa]*.

timezone|**tz** *tzname*
O nome do fuso horário do job. Consulte Capítulo 16, “Gerenciando fusos horários”, na página 597 para obter nomes válidos.

lowtime
Especifica o limite inferior de um intervalo de tempo, expresso no mesmo formato de *time*. São selecionadas apenas as tarefas que iniciaram nesta hora ou após ela.

hightime
Especifica o limite superior de um intervalo de tempo, expresso no mesmo formato de *time*. São selecionadas apenas as tarefas que iniciaram nesta hora ou antes dela.

Se **started** for usado sozinho e for precedido por +, então apenas as tarefas que começaram a ser executadas nessa hora serão selecionadas.

Se **started** for usado sozinho e for precedido por ~, então apenas as tarefas que começaram a executar nesse hora ou após ela e que continuam em execução serão selecionadas.

state=*state***[,...]**
Seleciona ou exclui jobs com base em seus estados.

state Especifica o estado atual do job. Os estados de job válidos são os seguintes:

ABEND
O job terminou com um código de saída diferente de zero.

ABENP
Uma confirmação **abend** foi recebida, mas o job não está concluído.

ADD A tarefa está sendo submetida.

CANCL
Apenas para dependências da interligação de redes. A tarefa remota ou o fluxo de tarefas foi cancelado.

DONE
A tarefa foi concluída em um estado desconhecido.

ERROR
Para dependências da interligação de redes apenas, ocorreu um erro durante a verificação do status remoto.

EXEC A tarefa está em execução. O sinalizador + ao lado do status EXEC significa que a tarefa é gerenciada pelo processo do **batchman** local.

EXTRN
O status é desconhecido apenas para as dependências da interligação de redes. Ocorreu um erro, uma ação reexecutar acabou de ser realizada na tarefa do fluxo de tarefas `EXTERNAL` ou a tarefa ou o fluxo de tarefas remoto não existe.

FAIL Impossível lançar a tarefa.

FENCE
O valor de prioridade do job está abaixo da cerca.

HOLD
A tarefa está aguardando a resolução da dependência.

INTRO
O job é apresentado para ser lançado pelo sistema. O sinalizador + ao lado do status INTRO significa que a tarefa é gerenciada pelo processo do **batchman** local.

PEND A tarefa foi concluída e está aguardando confirmação.

READY
A tarefa está pronta para ser lançada e todas as dependências foram resolvidas.

SCHED
A hora **at** da tarefa não chegou.

SUCC A tarefa foi concluída com um código de saída zero.

SUCCP
Uma confirmação **succ** foi recebida, mas o job não está concluído.

WAIT A tarefa está no estado WAIT (somente para o agente estendido).

until[=*time* **|** *lowtime***, |** *,hightime* **|** *lowtime***,***hightime* **]**
Seleciona ou exclui jobs com base em sua hora de término planejada.

time Especifica a hora de término planejada, expressa como a seguir:

hhmm[**+***n* **days** | *date*] [**timezone|tz** *tzname*]

hhmm A hora e minuto.

+*n* **days**
A próxima ocorrência de *hhmm* em *n* número de dias.

date A próxima ocorrência de *hhmm* em *date*, expressa como *mm/dd[/yy]*.

timezone|tz *tzname*
O nome do fuso horário do job. Consulte Capítulo 16, “Gerenciando fusos horários”, na página 597 para obter nomes válidos.

lowtime
Especifica o limite inferior de um intervalo de tempo, expresso no mesmo formato de *time*. São selecionados jobs que planejaram as horas de término nesta hora ou após ela.

hightime
Especifica o limite superior de um intervalo de tempo, expresso no mesmo formato de *time*. São selecionados jobs que planejaram as horas de término nesta hora ou antes dela.

Se **until** for utilizado sozinho e for precedido por +, então, as tarefas são selecionadas se tiverem uma hora **until** especificada.

Se **until** for utilizado sozinho e for precedido por ~, então, as tarefas são selecionadas se não tiverem uma hora **until** especificada.

Selecionando Fluxos de Tarefas em Comandos

Para comandos que operam em fluxos de job, os fluxos de job de destino são selecionados pela especificação de atributos e qualificadores.

Como ***scheddateandtime*** é especificado em minutos, a combinação do horário **jobstreamname** e do *scheddateandtime* pode não ser exclusiva. Por essa razão, o **jobstream_id** foi disponibilizado ao usuário, para finalidades de exibição ou para executar ações contra uma instância específica de um fluxo de tarefas.

A sintaxe de seleção do fluxo de tarefas é mostrada abaixo e descrita nas páginas a seguir. É possível escolher uma das duas sintaxes descritas.

Sintaxe

[*workstation***#**]*jobstreamname*(*hhmm*[*date*]) [{**+** | **~**}*jobstreamqualifier*[...]]

[*workstation***#**]*jobstream_id* **;schedid**

Argumentos

estação de trabalho
Especifica o nome da estação de trabalho na qual o fluxo de tarefa é executado. Exceto quando também utilizar `schedid`, caracteres curingas são permitidos.

jobstreamname
Especifica o nome do fluxo de tarefas. Caracteres curinga são permitidos.

(*hhmm* **[***date***])**
Indica a hora e a data que a instância do fluxo de tarefas é localizada no plano de pré-produção. Esse valor corresponde ao valor designado à palavra-chave **schedtime** na definição do fluxo de tarefas se nenhuma restrição de tempo **at** tiver sido configurada. Após a instância do fluxo de tarefas ter iniciado o processamento, o valor de ***hhmm* [*date*]** é configurado para a hora de início do fluxo de tarefas. A utilização de curingas nesse campo não é permitida. Ao emitir comandos **conman** sequenciais a partir do prompt de shell, coloque o comando **conman** entre aspas duplas **"**. Por exemplo, execute esse comando da seguinte forma:

```
conman "ss my_workstation#my_js(2101 02/23)"
```

jobstreamqualifier
Consulte "Qualificadores do Fluxo de Job" abaixo.

jobstream_id
Especifica o identificador do fluxo de tarefa alfanumérico exclusivo gerado automaticamente pelo planejador e designado para o fluxo de tarefa. É utilizado principalmente por processos internos para identificar essa instância do fluxo de tarefas no plano de produção, mas pode freqüentemente ser utilizado também ao gerenciar o fluxo de tarefas a partir do programa de linha de comandos **conman** especificando a opção **;schedid**.

schedid
Indica que o identificador do fluxo de tarefa é utilizado na seleção do fluxo de tarefa.

Nota: O Tivoli Workload Scheduler ajuda a identificar a instância correta do fluxo de tarefas quando a seleção de fluxo de tarefas fornece um resultado ambíguo se mais de uma instância atender seus critérios de seleção. Por exemplo, quando mais de uma instância de `WK1#J1` é incluída no plano de produção e, portanto, a seleção do fluxo de tarefas fornece um resultado ambíguo, o prompt a seguir é gerado automaticamente para permitir que você escolha a instância correta:

```
Process WK1#J1[(0600 03/04/06),(0AAAAAAAAAAAABTB)]
        (enter "y" for yes, "n" for no)?y
Command forwarded to batchman for WK1#J1[(0600 03/04/06),(0AAAAAAAAAAAABTB)]
Process WK1#J1[(1010 03/04/06),(0AAAAAAAAAAAABTC)]
        (enter "y" for yes, "n" for no)?n
```

Na saída, somente a instância do fluxo de tarefas planejada em `(0600 03/04/06)` e com o identificador `0AAAAAAAAAAAABTB` é selecionada para executar o comando.

Qualificadores de Fluxo de Tarefas

at[=*time* **|** *lowtime***, |** *,hightime* **|** *lowtime***,***hightime* **]**
Seleciona ou exclui fluxos de job com base na hora de início planejada.

time Especifica a hora de início planejada, expressa como a seguir:

hhmm[**+***n* **days** | *date*] [**timezone**|**tz** *tzname*]

hhmm A hora e minuto.

+*n* **days**
A próxima ocorrência de *hhmm* em *n* número de dias.

date A próxima ocorrência de *hhmm* em *date*, expressa como *mm/dd[/yy]*.

timezone | tz *tzname*
O nome do fuso horário do fluxo de job. Consulte Capítulo 16, “Gerenciando fusos horários”, na página 597 para obter nomes válidos.

lowtime
Especifica o limite inferior de um intervalo de tempo, expresso no mesmo formato de *time*. São selecionados fluxos de job que planejaram horas de início nesta hora ou após ela.

hightime
Especifica o limite superior de um intervalo de tempo, expresso no mesmo formato de *time*. São selecionados fluxos de job que planejaram horas de início nesta hora ou antes dela.

Se **at** for utilizado sozinho e for precedido por +, então, os fluxos de tarefa selecionados são aqueles que contêm uma dependência **at**.

Se **at** for utilizado sozinho e for precedido por ~, então, os fluxos de tarefa selecionados são aqueles que não contém uma dependência **at**.

carriedforward
Seleciona os fluxos de job que foram transportados se precedidos por +, exclui os fluxos de job que foram transportados se precedidos por ~.

carryforward
Se precedidos por +, seleciona os fluxos de job que foram planejados utilizando a palavra-chave **carryforward**; se precedidos por ~, exclui os fluxos de job que foram planejados utilizando a palavra-chave **carryforward**.

finished[=*time* | *lowtime***,** | **,***hightime* | *lowtime***,***hightime***]**
Seleciona ou exclui fluxos de job se tiverem terminado ou não.

time Especifica a hora exata em que os fluxos de job terminaram, expressa como a seguir:

hhmm [*date*] [**timezone | tz** *tzname*]

hhmm A hora e minuto.

date A próxima ocorrência de *hhmm* na data, expressa como *mm/dd[/aa]*.

timezone | tz *tzname*
O nome do fuso horário do fluxo de job. Consulte Capítulo 16, “Gerenciando fusos horários”, na página 597 para obter nomes válidos.

lowtime
Especifica o limite inferior de um intervalo de tempo, expresso no mesmo formato de *time*. São selecionados fluxos de job que terminaram nessa hora ou após ela.

hightime
Especifica o limite superior de um intervalo de tempo, expresso no mesmo formato de *time*. São selecionados fluxos de job que terminaram nessa hora ou antes dela.

Se **finished** for utilizado sozinho e estiver precedido por +, os fluxos de job selecionados serão os jobs cuja execução terminou.

Se **finished** for utilizado sozinho e estiver precedido por ~, os fluxos de job selecionados serão os jobs cuja execução não terminou.

follows=[*netagent***::][***workstation***#]{***jobstreamname***(***hhmm* **[***mm/dd***[/***yy***]])[.***job* | **@] |** *jobstream_id***.***job***;schedid}|** *job***[;nocheck] [,...]**

Seleciona ou exclui fluxos de tarefas se tiverem ou não uma dependência `follows`.

netagent

Especifica o nome do agente de rede que faz a interface com a rede remota do Tivoli Workload Scheduler que contém a tarefa ou o fluxo de tarefas pré-requisito. Caracteres curinga são permitidos. Para obter informações adicionais sobre agentes de rede, consulte Capítulo 18, "Gerenciando Dependências da Interligação de Redes", na página 615.

estação de trabalho

Especifica o nome da estação de trabalho na qual o job ou fluxo de job de pré-requisito é executado. Caracteres curinga são permitidos.

jobstreamname

Especifica o nome do fluxo de tarefas de pré-requisito. Caracteres curinga são permitidos.

jobname

Especifica o nome do job de pré-requisito. Caracteres curinga são permitidos.

jobstream_id

Especifica o identificador do fluxo de tarefa exclusivo. Consulte "Argumentos" na página 354 para obter informações adicionais sobre identificadores de fluxo de tarefas.

schedid

Essa palavra-chave, se presente, aplica-se a todos os identificadores de fluxos de tarefas especificados na cláusula e indica que, para todos os fluxos de tarefas especificados, você está utilizando os *jobstream_id*s, não os *jobstreamname*s. Se quiser selecionar alguns fluxos de tarefas utilizando o *jobstream_id* e alguns fluxos de tarefas utilizando o *jobstreamname*, você deverá utilizar duas cláusulas **follows** diferentes, uma contendo os fluxos de tarefas identificados pelas palavras-chave *jobstreamname* sem as palavras-chave **schedid** e a outra contendo os fluxos de tarefas identificados pelo *jobstream_id* com a palavra-chave **schedid**.

nocheck

É válido somente para comandos de envio e utilizado em conjunto com a palavra-chave **schedid**. A palavra-chave **nocheck** indica que **conman** não precisa verificar a existência da tarefa pré-requisito *jobstream_id.job* no arquivo `Symphony`. Supõe-se que *jobstream_id.job* exista. Caso não exista, **batchman** imprimirá uma mensagem de aviso em `stdlist`.

Se **follows** for utilizado sozinho e estiver precedido por +, os fluxos de job serão selecionados se contiverem dependências `follows`.

Se **follows** for utilizado sozinho e estiver precedido por ~, os fluxos de job serão selecionados se não contiverem dependências `follows`.

limit[=*limit* **|** *lowlimit***, | ,***highlimit* **|** *lowlimit***,***highlimit***]**
Seleciona ou exclui fluxos de job se tiverem ou não um limite de job.

limite Especifica o valor exato do limite de job.

lowlimit
Especifica o limite inferior de faixa. São selecionados fluxos de job que possuem limites de job iguais ou maiores que esse limite.

highlimit
Especifica o limite superior de uma faixa. São selecionados fluxos de job que possuem limites de job menores ou iguais a esse limite.

Se **limit** for utilizado sozinho e estiver precedido por +, os fluxos de job serão selecionados se tiverem qualquer limite de job.

Se **limit** for utilizado sozinho e estiver precedido por ~, os fluxos de job serão selecionados se não tiverem nenhum limite de job.

needs[=[*workstation***#]***resourcename***]**
Seleciona ou exclui fluxos de job se tiverem ou não uma dependência de recurso.

estação de trabalho
Especifica o nome da estação de trabalho na qual o recurso foi definido. Caracteres curinga são permitidos.

resourcename
Especifica o nome do recurso. Caracteres curinga são permitidos.

Se **needs** for utilizado sozinho e estiver precedido por +, os fluxos de job serão selecionados se contiverem dependências `resource`.

Se **needs** for utilizado sozinho e estiver precedido por ~, os fluxos de job serão selecionados se não contiverem dependências `resource`.

opens[=[*workstation***#]***filename***[(***qualifier***)]]**
Seleciona ou exclui fluxos de job se tiverem ou não uma dependência de arquivo. Uma dependência file ocorre quando um job ou fluxo de job depende da existência de um ou mais arquivos para poder começar a execução.

estação de trabalho
Especifica o nome da estação de trabalho na qual o arquivo existe. Caracteres curinga são permitidos.

nome do arquivo
Especifica o nome do arquivo. O nome deverá ser colocado entre aspas (") se contiver caracteres especiais diferentes do seguinte: alfanuméricos, traços (-), barras (/), barras invertidas (\) e sublinhados (_). Caracteres curinga são permitidos.

qualifier
Uma condição de teste válida. Se omitido, os fluxos de job serão selecionados ou excluídos independente de um qualificador.

Se **opens** for utilizado sozinho e estiver precedido por +, os fluxos de job serão selecionados se contiverem dependências `file`.

Se **opens** for utilizado sozinho e estiver precedido por ~, os fluxos de job serão selecionados se não contiverem dependências `file`.

priority=*pri* **|** *lowpri***, | ,***highpri* **|** *lowpri***,***highpri*
Seleciona ou exclui fluxos de job com base em suas prioridades.

pri Especifica o valor de prioridade. Você pode digitar de **0** a **99**, **hi** ou **go**.

lowpri Especifica o limite inferior de uma faixa de prioridades. Os fluxos de job são selecionados com prioridades iguais ou maiores que esse valor.

highpri Especifica o limite superior de uma faixa de prioridades. Os fluxos de job são selecionados com prioridades menores ou iguais a esse valor.

prompt[=*promptname* | *msgnum***]**
Seleciona ou exclui fluxos de job se tiverem ou não uma dependência de prompt.

promptname
Especifica o nome de um prompt global. Caracteres curinga são permitidos.

msgnum
Especifica o número de mensagem de um prompt local.

Se **prompt** for utilizado sozinho e estiver precedido por +, os fluxos de job serão selecionados se contiverem dependências `prompt`.

Se **prompt** for utilizado sozinho e estiver precedido por ~, os fluxos de job serão selecionados se não contiverem dependências `prompt`.

started[=*time* | *lowtime*, | ,*hightime* | *lowtime*,*hightime***]**
Seleciona ou exclui fluxos de job se tiverem ou não sido iniciados.

time Especifica a hora exata em que o fluxo de job foi iniciado, expressa como a seguir:

hhmm [*date*] [**timezone**|**tz** *tzname*]

hhmm A hora e minuto.

date A próxima ocorrência de *hhmm* na data, expressa como *mm/dd[/aa]*.

timezone|**tz** *tzname*
O nome do fuso horário do fluxo de job. Consulte Capítulo 16, “Gerenciando fusos horários”, na página 597 para obter nomes válidos.

lowtime
Especifica o limite inferior de um intervalo de tempo, expresso no mesmo formato de *time*. São selecionados fluxos de job que iniciaram nessa hora ou após ela.

hightime
Especifica o limite superior de um intervalo de tempo, expresso no mesmo formato de *time*. São selecionados fluxos de job que iniciaram nessa hora ou antes dela.

Se **started** for utilizado sozinho e estiver precedido por +, os fluxos de job selecionados serão os jobs cuja execução foi iniciada.

Se **started** for utilizado sozinho e estiver precedido por ~, os fluxos de job selecionados serão os jobs cuja execução não foi iniciada.

state=*state***[,...]**
Seleciona ou exclui fluxos de job com base em seus estados.

state Especifica o estado atual do fluxo de job. Os estados válidos dos fluxos de job são os seguintes:

ABEND
O fluxo de job terminou de forma anormal.

ADD O fluxo de tarefas acaba de ser enviado.

EXEC O fluxo de tarefas está em execução.

EXTRN
Apenas para dependências da interligação de redes. Este é o estado do fluxo de tarefas EXTERNAL contendo tarefas que referenciam tarefas ou fluxos de tarefas na rede remota.

HOLD
O fluxo de tarefa está aguardando a resolução de dependências.

READY
O fluxo de job está pronto para ser ativado e todas as dependências foram resolvidas.

STUCK
A execução foi interrompida. Nenhum job será lançado sem a intervenção do operador.

SUCC O fluxo de job foi concluído com êxito.

until[=*time* **|** *lowtime***, | ,***hightime* **|** *lowtime***,***hightime* **]**
Seleciona ou exclui fluxos de job com base no horário de encerramento planejado.

time Especifica a hora de término planejada, expressa como a seguir:

hhmm[**+***n* **days** | *date*] [**timezone|tz** *tzname*]

hhmm A hora e minuto.

+*n* **days**
A próxima ocorrência de *hhmm* em *n* número de dias.

date A próxima ocorrência de *hhmm* em *date*, expressa como *mm/dd[/yy]*.

timezone|tz *tzname*
O nome do fuso horário do fluxo de job. Consulte Capítulo 16, “Gerenciando fusos horários”, na página 597 para obter nomes válidos.

lowtime
Especifica o limite inferior de um intervalo de tempo, expresso no mesmo formato de *time*. São selecionados fluxos de job que planejaram horas de término nessa hora ou após ela.

hightime
Especifica o limite superior de um intervalo de tempo, expresso no mesmo formato de *time*. São selecionados fluxos de job que planejaram horas de término nessa hora ou antes dela.

Se **until** for utilizado sozinho e estiver precedido por +, os fluxos de job selecionados serão aqueles que contêm o horário de encerramento planejado.

Se **until** for utilizado sozinho e estiver precedido por ~, os fluxos de job selecionados serão aqueles que não contêm o horário de encerramento planejado.

Gerenciando Tarefas e Fluxos de Tarefas a partir de Agentes de Nível Anterior

A alteração na convenção de nomenclatura da instância do fluxo de tarefas introduzida com o Tivoli Workload Scheduler versão 8.3 requer que você aplique a seguinte solução alternativa ao emitir comandos da linha de comandos em um plano gerado em um gerenciador de domínio principal do Tivoli Workload Scheduler versão 8.3 (ou posterior) a partir dos agentes do Tivoli Workload Scheduler versão 8.1, 8.2 ou 8.2.1:

- Você deve utilizar o símbolo **@** como o primeiro caractere para o identificador da instância do fluxo de tarefas. Por exemplo, o fluxo de tarefas em execução na estação de trabalho `CPU1` com o identificador `0AAAAAAAAAAAAY3` deve ser identificado na linha de comandos **conman** conforme a seguir:

  ```
  CPU1#@AAAAAAAAAAAY3
  ```

- Você não pode utilizar a palavra-chave **follows** ao incluir uma dependência em uma tarefa ou em um fluxo de tarefas ou ao enviar como uma tarefa um comando ou um arquivo.
- Você não pode utilizar a palavra-chave **into** para especificar o fluxo de tarefa no qual a tarefa deve ser incluída ao enviar como uma tarefa um comando ou um arquivo.

Por exemplo, para exibir as informações sobre a tarefa `job2` contida na instância do fluxo de tarefas com o identificador `0AAAAAAAAAAAAT1` em execução na estação de trabalho `CPU1`, execute o seguinte comando nos agentes do Tivoli Workload Scheduler versão 8.1, 8.2 ou 8.2.1:

```
sj CPU1#@AAAAAAAAAAAAT1.job2
```

Estas alterações também serão vistas nos relatórios e logs e em qualquer outro local no qual os nomes do fluxo de tarefas forem impressos ou exibidos.

Conman commands

A Tabela 65 lista os comandos **conman**. Os nomes e as palavras-chave dos comandos podem ser digitados em caracteres maiúsculos ou minúsculos e podem ser abreviados com o mínimo de caracteres iniciais, o necessário para distingui-los exclusivamente uns dos outros. Alguns dos nomes dos comandos também têm formas curtas específicas.

Nota: Os tipos de estações de trabalho na tabela a seguir tem estes significados:

- D - gerenciadores de domínio dinâmico, gerenciadores de domínio de backup
- M - gerenciadores de domínios principais e principais de backup
- F - gerenciadores de domínios e agentes tolerantes a falhas
- T - para agentes tolerantes a falha
- S - agentes padrão (é possível exibir arquivos somente em um agente padrão)

Tabela 65. Lista de Comandos do Conman

Comando	Forma Abreviada	Descrição	Tipo	Página
adddep { job | sched }	**adj | ads**	Adiciona dependências de job ou fluxo de job.	D	“adddep job” na página 362 “adddep sched” na página 364
altpass		Altera a senha de uma definição de objeto do usuário.	D	“altpass” na página 366

Tabela 65. Lista de Comandos do Conman (continuação)

Comando	Forma Abreviada	Descrição	Tipo	Página
altpri	**ap**	Altera prioridades de job ou fluxo de job.	D	"altpri" na página 367
bulk_discovery	**bulk**	Desempenha uma descoberta em massa. Para utilizar com o IBM Tivoli Monitoring 6.1 (Tivoli Enterprise Portal).	D	"bulk_discovery" na página 368
cancel { job | sched }	**cj | cs**	Cancela um job ou fluxo de job.	D	"cancel job" na página 368 "cancel sched" na página 370
checkhealthstatus	**chs**	Chama o serviço **chkhltst** para verificar se a caixa de correio pode ser lida com sucesso por **mailman** ou se há erros no cabeçalho da caixa de correio.	MFS	"checkhealthstatus" na página 371
confirmar	**conf**	Confirma conclusão do job.	D	"confirm" na página 372
console	**cons**	Atribui o console do Tivoli Workload Scheduler.	FS	"console" na página 373
continue	**cont**	Ignora o próximo erro.	FS	"continue" na página 374
deldep { job | sched }	**ddj | dds**	Elimina dependências de job ou fluxo de job.	D	"deldep job" na página 374 "deldep sched" na página 376
deployconf	**deploy**	Obtém a configuração de monitoramento mais recente para o mecanismo de monitoramento de eventos na estação de trabalho.	FS	"deployconf" na página 377
display { file | job | sched }	**df | dj | ds**	Exibe arquivos, jobs e fluxos de job.	FS	"display" na página 378
exit	**e**	Sai do **conman**.	FS	"exit" na página 380
fence	**A**	Define o limitador de job do Tivoli Workload Scheduler.	D	"fence" na página 380
help(1)	**h**	Exibe informações sobre o comando.	FS	"help" na página 382
kill	**k**	Pára um job em execução.	D	"kill" na página 383
limit { cpu | sched }	**lc | ls**	Altera o limite de job de uma estação de trabalho ou fluxo de job.	D	"limit cpu" na página 384 "limit sched" na página 385
link	**lk**	Abre links da estação de trabalho.	FS	"link" na página 386
listsym	**lis**	Exibe uma lista de arquivos de log do `Symphony`.	D	"listsym" na página 388
recall	**rc**	Exibe mensagens de prompt.	D	"recall" na página 390
redo	**red**	Edita o comando anterior.	FS	"redo" na página 391
release { job | sched }	**rj | rs**	Libera dependências de job ou de fluxo de job.	D	"release job" na página 392 "release sched" na página 393
reply	**rep**	Responde mensagens de prompts.	D	"reply" na página 395
rerun	**rr**	executa novamente uma tarefa.	D	"rerun" na página 396
resetFTA	**N/D**	Recupera ujm arquivo Symphony corrompido no agente tolerante a falhas especificado	P	"resetFTA" na página 398
resource	**res**	Altera o número de unidades de recurso.	D	"resource" na página 399
setsym	**set**	Seleciona um arquivo de log do Symphony.	D	"setsym" na página 400
showcpus	**sc**	Exibe informações de links e de estação de trabalho.	FS	"showcpus" na página 401
showdomain	**showd**	Exibe informações sobre o domínio.	FS	"showdomain" na página 408
showfiles	**sf**	Exibe informações sobre arquivos.	D	"showfiles" na página 409
showjobs	**sj**	Exibe informações sobre jobs.	D	"showjobs" na página 411
showprompts	**sp**	Exibe informações sobre prompts.	D	"showprompts" na página 427
showresources	**sr**	Exibe informações sobre recursos.	D	"showresources" na página 429
showschedules	**ss**	Exibe informações sobre fluxos de job.	D	"showschedules" na página 431
shutdown	**shut**	Para processos de produção do Tivoli Workload Scheduler.	FS	"shutdown" na página 436
start		Inicia processos de produção do Tivoli Workload Scheduler.	FS	"start" na página 437
startappserver		Inicia o processo do WebSphere Application Server	DM	"startappserver" na página 439
startbrokerapp		Inicia o aplicativo dynamic workload broker.	DM	"startbrokerapp" na página 440
starteventprocessor	**startevtp**	Inicia o servidor de processamento de eventos.	M(2)	"starteventprocessor" na página 440
startmon	**startm**	Inicia o processo `monman` que ativa o mecanismo de monitoramento de eventos no agente.	FS	"startmon" na página 441
status	**stat**	Exibe o status de produção do Tivoli Workload Scheduler.	FS	"status" na página 442
parar		Para processos de produção do Tivoli Workload Scheduler.	FS	"stop" na página 442
stop ;progressive		Pára os processos de produção do Tivoli Workload Scheduler hierarquicamente.		"stop ;progressive" na página 444
stopappserver	**stopapps**	Para o processo do WebSphere Application Server	DM	"stopappserver" na página 445
stopbrokerapp		Para o aplicativo dynamic workload broker.	DM	"stopbrokerapp" na página 447
stopeventprocessor	**stopevtp**	Pára o servidor de processamento de eventos.	M(2)	"stopeventprocessor" na página 447

Tabela 65. Lista de Comandos do Conman (continuação)

Comando	Forma Abreviada	Descrição	Tipo	Página
stopmon	**stopm**	Pára o mecanismo de monitoramento de eventos no agente.	FS	"stopmon" na página 448
submit { docommand | file | job | sched }	**sbd | sbf | sbj | sbs**	Submete um comando, arquivo, job ou fluxo de job.	FS([3])	"submit docommand" na página 449 "submit file" na página 452 "submit job" na página 456 "submit sched" na página 459
switcheventprocessor	**switchevtp**	Comuta o servidor de processamento de eventos de gerenciadores de domínio principal para principais de backup ou vice-versa.	S	"switcheventprocessor" na página 463
switchmgr	**switchm**	Alterna o gerenciador de domínio.	D	"switchmgr" na página 464
system-command		Envia um comando ao sistema.	FS	"system command" na página 465
tellop	**to**	Envia uma mensagem ao console.	FS	"tellop" na página 466
unlink		Fecha links da estação de trabalho.	FS	"unlink" na página 466
versão	**v**	Exibe o banner d programa de linha de comandos de **conman**.	FS	"version" na página 469

1. Não disponível no sistema operacional Windows suportado.
2. Inclui estações de trabalho instaladas como principais de backup mas utilizadas como agente tolerante a falhass comuns.
3. É possível utilizar **submit job** (**sbj**) e **submit sched** (**sbs**) em um agente padrão, utilizando os parâmetros de conexão ou especificando as configurações no arquivo `useropts` ao chamar a linha de comandos **conman**.

Nota: Nos comandos, os termos *sched* e *schedule* referem-se a *fluxos de job* e o termo *CPU* refere-se a *estações de trabalho*.

adddep job

Adiciona dependências a um job.

Você deve ter acesso ***adddep*** ao job. Para incluir dependências `needs` e `prompt`, você deve ter acesso ***use*** aos recursos e prompts globais.

Sintaxe

{**adddep job** | **adj**} *jobselect*
;*dependency*[**;**...]
[**;noask**]

Argumentos

jobselect
Consulte "Selecionando Tarefas em Comandos" na página 345.

dependência
O tipo de dependência. Especifique uma das seguintes. Caracteres curinga não são permitidos.

at=*hhmm*[**timezone** | **tz** *tzname*][**+***n* **days** | *mm/dd[/yy]*] | [**absolute** | **abs**]

confirmed

deadline=*time* [**timezone**|**tz** *tzname*][**+***n* **day**[**s** | *mm/dd[/yy]*]

every=*rate*

follows=[*netagent***::**][*workstation***#**]{*jobstreamname*[*hhmm* [*mm/dd[/yy]*]][*.job* | **@**] | *jobstream_id.job***;schedid**}| *job*[**,**...]

maxdur=[*hhhmm*] [**onmaxdur** *action*]

mindur=[*hhhmm*] [**onmindur** *action*]

needs=[*num*] [*workstation***#**]*resource*[**,**...]

opens=[*workstation***#**]"*filename*"[**(***qualifier***)**][**,**...] **priority**[**=***pri* | **hi** | **go**]

prompt="[**:** | **!**]*text***"** | *promptname*[**,**...]

until *time* [**timezone** | **tz** *tzname*][**+***n* **day**[**s**]] | [**absolute** | **abs**] [**;onuntil** *action*]

noask Especifica para não solicitar confirmação antes da tomada de ação em cada job qualificado.

Nota:

1. Se você incluir uma dependência duas vezes em um fluxo de tarefa para uma tarefa, ambas as dependências serão tratadas.
2. Ao usar a palavra-chave **prazo final**, certifique-se de que a opção **bm check deadline** esteja configurada para um valor maior que 0 no arquivo de configuração `localopts` na estação de trabalho em que você estiver trabalhando. É possível definir a opção **bm check deadline** em cada estação de trabalho na qual você deseja estar ciente sobre a expiração do prazo final, ou, se você desejar obter informações atualizadas sobre o ambiente todo, defina a opção no gerenciador de domínio principal. Prazos finais para tarefas críticas são avaliadas automaticamente, independentemente da opção **bm check deadline**. Para obter mais informações sobre a opção **bm check deadline**, consulte Detalhes das Opções Locais.

Comentários

Se você não especificar um valor para `priority`, o job reverterá para sua prioridade planejada original. Se você não especificar uma estação de trabalho em `follows`, `needs` ou `opens`, o padrão será a estação de trabalho na qual o job é executado.

Não é possível utilizar esse comando para incluir `resource` ou `prompt` como dependência, a não ser que já estejam referenciadas por uma tarefa ou um fluxo de tarefas no arquivo `Symphony`.

Exemplos

Para incluir uma dependência de recurso na tarefa `job3` no fluxo de tarefas `sked9(0900 02/19/06)`, execute o seguinte comando:

```
adj sked9(0900 02/19/06).job3 ; needs=2 tapes
```

Para incluir uma dependência follows externa a partir da tarefa `JOB022` no fluxo de tarefas `MLN#SCHED_02(0600 02/24)` em `JOBA` no fluxo de tarefas `MLN#NEW_TEST(0900 02/19/06)`, execute o seguinte comando:

```
adj MLN#NEW_TEST(0900 02/19/06).JOBA ; follows MLN#SCHED_02(0600 02/24/06).JOB022
```

Para incluir uma dependência de arquivo e uma hora **until** na tarefa `j6` no fluxo de tarefas `JS2(0900 02/19/06)`, execute o seguinte comando:

```
adj WK1#JS2(0900 02/19/06).j6 ; opens="/usr/lib/prdata/file5"(-s %p) ; until=2330
```

Para eliminar uma tarefa quando ela tiver sido executada por mais de 9 horas e 1 minuto, execute o seguinte comando:

```
conman addep BROOKS#JOBDEF1 ;maxdur=901 ;onmaxdur kill
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Monitorar Tarefas**.
2. Selecione **Todas as Tarefas no plano** ou outra tarefa para monitorar tarefas.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. Na tabela que contém a lista de tarefas, selecione a tarefa na qual você deseja incluir uma dependência e clique na guia **Dependências...**
5. No painel Dependências, expanda uma seção de dependência e clique no botão correspondente à ação você deseja executar.

adddep sched

Adiciona dependências a um fluxo de tarefa.

Você deve ter acesso ***adddep*** ao fluxo de tarefa. Para incluir dependências `needs` e `prompt`, você deve ter acesso ***use*** aos recursos e prompts globais.

Sintaxe

{**adddep sched** | **ads**} *jstreamselect*
;*dependency*[**;**...]
[**;noask**]

Argumentos

jstreamselect
Consulte "Selecionando Fluxos de Tarefas em Comandos" na página 353.

dependência
O tipo de dependência. Especifique uma das seguintes. Caracteres curinga não são permitidos.

at=*hhmm*[**timezone** | **tz** *tzname*][**+***n* **days** | *mm/dd[/yy]*] | [**absolute** | **abs**]

carryforward

deadline=*time* [**timezone**|**tz** *tzname*][**+***n* **day**[**s** | *mm/dd[/yy]*]

follows=[*netagent***::**][*workstation***#**]{*jobstreamname*[*hhmm* [*mm/dd[/yy]*]][*.job* | **@**] | *jobstream_id.job***;schedid**}| *job*[**,**...]

limit=*limit*

needs=[*num*] [*workstation***#**]*resource*[**,**...]

opens=[*workstation***#**]"*filename*"[**(***qualifier***)**][**,**...] **priority**[**=***pri* | **hi** | **go**]

prompt="[**:** | **!**]*text***"** | *promptname*[**,**...]

until *time* [**timezone**|**tz** *tzname*][**+***n* **day**[**s**] | [**absolute** | **abs**]] [**;onuntil** *action*]

noask Especifica para não solicitar confirmação antes da tomada de ação em cada fluxo de job qualificado.

Nota:

1. Se você incluir duas vezes uma dependência em um fluxo de tarefas para outro fluxo de tarefas, somente uma dependência será considerada.
2. Ao usar a palavra-chave **prazo final**, certifique-se de que a opção **bm check deadline** esteja configurada para um valor maior que 0 no arquivo de configuração `localopts` na estação de trabalho em que você estiver trabalhando. É possível definir a opção **bm check deadline** em cada estação de trabalho na qual você deseja estar ciente sobre a expiração do prazo final, ou, se você desejar obter informações atualizadas sobre o ambiente todo, defina a opção no gerenciador de domínio principal. Prazos finais para tarefas críticas são avaliadas automaticamente, independentemente da opção **bm check deadline**. Para obter mais informações sobre a opção **bm check deadline**, consulte Detalhes das Opções Locais.

Comentários

- Se você não especificar um valor para `priority`, o fluxo de tarefa reverterá para sua prioridade planejada original.
- Se você não especificar um valor para **limit**, o valor assumirá o padrão de 0.
- Se você não especificar uma estação de trabalho em `follows`, `needs` ou `opens`, o padrão será a estação de trabalho na qual o fluxo de tarefa é executado.
- Não é possível utilizar esse comando para incluir um `resource` ou um `prompt` como dependências, a menos que já existam no plano de produção. Para ver qual recurso e prompts existem no plano, consulte “showresources” na página 429 e “showprompts” na página 427.

Exemplos

Para incluir uma dependência de prompt no fluxo de tarefas `sked9(0900 02/19/06)`, execute o seguinte comando:

```
ads sked9(0900 02/19/06) ; prompt=msg103
```

Para incluir uma dependência follows externa a partir da tarefa `JOBB` no fluxo de tarefas `CPUA#SCHED_02(0600 02/24/06)` e um limite de tarefa no fluxo de tarefas `CPUA#TEST(0900 02/19/06)`, execute o seguinte comando:

```
ads CPUA#TEST(0900 02/19/06) ; follows CPUA#SCHED_02(0600 02/24/06).JOBB; limit=2
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Monitorar Fluxos de Tarefas**.
2. Selecione **Todos os Fluxos de tarefas no plano** ou outra tarefa para monitorar fluxos de tarefas.
3. Escolha um nome de mecanismo ou especifique as propriedades da conexão e clique em **OK**
4. Na tabela que contém a lista de fluxos de tarefas, selecione o fluxo de tarefas no qual deseja incluir uma dependência e clique na guia **Dependências...**
5. No painel Dependências, expanda uma seção de dependência e clique no botão correspondente à ação você deseja executar.

altpass

Altera a senha de um objeto de usuário no plano de produção atual.

Você deve ter acesso ***altpass*** ao objeto do usuário.

Sintaxe

altpass
[*workstation***#**]
username
[**;"***password***"**]

Argumentos

estação de trabalho
Especifica a estação de trabalho na qual o usuário está definido. Use maiúsculas para este campo mesmo apesar de ter usado letras maiúsculas e minúsculas ao especificar a *estação de trabalho* na definição do usuário. Para obter informações adicionais, consulte "Definição do Usuário" na página 192. Não especifique este campo se o usuário pertencer a um domínio do Windows gerenciado pelo diretório ativo. O padrão é a estação de trabalho na qual o **conman**está sendo executado.

username
Especifica o nome de um usuário. Use o mesmo nome de usuário especificado no banco de dados do Tivoli Workload Scheduler e observe que eles fazem distinção entre maiúsculas e minúsculas. Para obter informações adicionais, consulte "Definição do Usuário" na página 192.

password
Especifica a nova senha. Ela deve ser colocada entre aspas duplas. Para indicar ausência de senha para o usuário, use duas aspas duplas consecutivas (**""**).

Comentários

Se você não especificar uma *senha*, o **conman** solicitará uma e uma confirmação. A senha não será exibida durante a digitação e não deverá ser colocada entre aspas. Observe que a alteração é feita somente no plano de produção atual, sendo, desse modo, temporária. Para fazer uma mudança permanente, consulte "Definição do Usuário" na página 192.

Exemplos

Para alterar a senha do usuário `Jim` na estação de trabalho `mis5` para `mynewpw`, execute o seguinte comando:

```
altpass MIS5#JIM;"mynewpw"
```

Para alterar a senha do usuário `jim` na estação de trabalho `Mis5` para `mynewpw` sem exibir a senha, execute o seguinte comando:

```
altpass MIS5#JIM
password: xxxxxxxx
confirm: xxxxxxxx
```

Para alterar a senha do usuário `Jim`, definida em um domínio do Windows gerenciador por diretório ativo denominado `twsDom`, para `mynewpw`, execute o seguinte comando:

```
altpass TWSDOM\JIM;"mynewpw"
```

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

o Guia do Usuário do Console do Dynamic Workload, seção Criando uma Senha de Usuário no Plano.

altpri

Altera a prioridade de um job ou fluxo de tarefa.

Você deve ter acesso ***altpri*** ao job ou fluxo de tarefa.

Sintaxe

{**altpri** | **ap**} *jobselect* | *jstreamselect*
[**;***pri*]
[**;noask**]

Argumentos

jobselect
Consulte "Selecionando Tarefas em Comandos" na página 345.

jstreamselect
Consulte "Selecionando Fluxos de Tarefas em Comandos" na página 353.

pri Especifica o nível de prioridade. É possível digitar um valor de **0** a **99**, **hi** ou **go**.

noask Especifica para não solicitar confirmação antes da tomada de ação em cada tarefa ou fluxo de tarefa qualificado.

Exemplos

Para alterar a prioridade da tarefa `balance` no fluxo de tarefas `glmonth(0900 02/19/06)`, execute o seguinte comando:

```
ap glmonth(0900 02/19/06).balance;55
```

Para alterar a prioridade do fluxo de tarefas `glmonth(0900 02/19/06)`, execute o seguinte comando:

```
ap glmonth(0900 02/19/06);10
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Monitorar Tarefas** ou **Monitorar Fluxos de Tarefas**.
2. Selecione uma tarefa de monitor.
3. Escolha um nome de mecanismo ou especifique as propriedades da conexão e clique em **OK**
4. Na tabela que contém a lista de objetos, selecione a tarefa ou fluxo de tarefas cuja prioridade você deseja alterar e clique na guia **Mais Ações** > **Prioridade**.
5. No painel exibido, configure a nova prioridade e clique em **OK**.

bulk_discovery

Solicita uma descoberta em massa para atualizar o status atual dos objetos monitorados. É usado para integração com o IBM Tivoli Monitoring 6.1 (Tivoli Enterprise Portal).

Você deve ter o acesso ***display*** para o objeto do arquivo.

Sintaxe

{**bulk_discovery** | **bulk**}

Comentários

Quando a integração com o IBM Tivoli Monitoring 6.1 está ativada, o comando **bulk_discovery** verifica o status de todas as tarefas monitoradas e fluxos de tarefas no plano e grava os eventos correspondentes no arquivo do log de eventos.

Por padrão, os eventos são gravados no arquivo **event.log**.

As mensagens que indicam o início e o encerramento da atividade de descoberta em massa são registradas no arquivo `twsmerge.log`.

cancel job

Cancela um job.

Você deve ter acesso ***cancel*** ao job.

Sintaxe

{**cancel job** | **cj**} *jobselect*
[;pend]
[;noask]

Argumentos

jobselect
Consulte "Selecionando Tarefas em Comandos" na página 345.

pend Cancela o job somente após a resolução de suas dependências.

noask Especifica para não solicitar confirmação antes da tomada de ação em cada job qualificado.

Comentários

Se você cancelar um job antes de ser ativado, ele não será ativado. Se você cancelar um job depois de ser ativado, sua execução continuará. Se você cancelar um job que estiver em execução e ele for concluído no estado ABEND, não serão tentadas etapas automáticas de recuperação do job.

Se você não utilizar a opção **;pend**, os jobs e fluxos de job dependentes do job cancelado serão liberados imediatamente da dependência.

Se você incluir a opção **;pend** e o job não tiver sido lançado, o cancelamento será deferido até que todas as dependências, incluindo uma hora **at**, sejam resolvidas. Depois de todas as dependências terem sido resolvidas, o job será cancelado e

todos os jobs ou fluxos de job dependentes do job cancelado serão liberados da dependência. Durante o período em que o cancelamento é deferido, a notação **[Cancel Pend]** é listada na coluna Dependências do job em uma tela **showjobs**.

Se você incluir a opção **;pend** e o job já tiver sido lançado, a opção será ignorada e todos os jobs ou fluxos de job dependentes do job cancelado serão liberados imediatamente da dependência.

É possível utilizar o comando **rerun** para executar novamente os jobs que foram cancelados ou que estiverem marcados com **[Cancel Pend]**. Também pode adicionar e excluir dependências de jobs que estiverem marcados com **[Cancel Pend]**.

Para cancelar imediatamente um job marcado com **[Cancel Pend]**, é possível digitar um comando **release** para o job ou digitar outro comando **cancel** sem a opção **;pend**.

Para jobs com horas **until** expiradas, a notação **[Until]** é listada na coluna Dependências em uma tela **showjobs** e suas dependências não são mais avaliadas. Se esse job também estiver marcado com **[Cancel Pend]**, ele não será cancelado até que você libere ou exclua a hora **until**, ou digite outro comando **cancel** sem a opção **;pend**.

Para parar de avaliar as dependências, defina a prioridade de um job como zero com o comando **altpri**. Para continuar a avaliá-las, defina a prioridade com um valor maior que zero.

Nota: No caso de dependências da interligação de redes, cancelar uma tarefa no fluxo de tarefas `EXTERNAL` liberará todas as tarefas e fluxos de tarefas locais da dependência. As tarefas do fluxo de tarefas `EXTERNAL` representam tarefas e fluxos de tarefas que foram especificados como dependências da interligação de redes. O status de uma dependência entre redes não é verificado após um comando **cancel** ser executado. Para obter informações adicionais, consulte "Gerenciando Dependências da Interligação de Redes no Plano" na página 620.

Exemplos

Para cancelar a tarefa `report` no fluxo de tarefas `apwkly(0900 02/19/06)` na estação de trabalho `site3`, execute o seguinte comando:

```
cj site3#apwkly(0900 02/19/06).report
```

Para cancelar a tarefa `setup` no fluxo de tarefas `mis5(1100 02/10/06)`, se não estiver no estado ABEND, execute o seguinte comando:

```
cj mis5(1100 02/10/06).setup~state=abend
```

Para cancelar a tarefa `job3` no fluxo de tarefas `sked3(0900 02/19/03)` somente após a resolução de suas dependências, execute o seguinte comando:

```
cj sked3(0900 02/19/06).job3;pend
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Monitorar Tarefas**.
2. Selecione **Todas as Tarefas no plano** ou outra tarefa para monitorar tarefas.

3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. Na tabela que contém a lista de tarefas, selecione uma tarefa e clique em **Mais Ações** > **Cancelar**.

cancel sched

Cancela um fluxo de job.

Você deve ter acesso ***cancel*** ao fluxo de job.

Sintaxe

{**cancel sched** | **cs**} *jstreamselect*
[**;pend**]
[**;noask**]

Argumentos

jstreamselect
Consulte "Selecionando Fluxos de Tarefas em Comandos" na página 353.

pend Cancela o fluxo de job somente após a resolução de suas dependências.

noask Especifica para não solicitar confirmação antes da tomada de ação em cada fluxo de job qualificado.

Comentários

Se você cancelar um fluxo de job antes de ele ser ativado, ele não será ativado. Se você cancelar um fluxo de job depois de ele ser ativado, os jobs iniciados serão concluídos, mas nenhum outro será ativado.

Se você não utilizar a opção **;pend**, os jobs e fluxos de job dependentes do fluxo de job cancelado serão liberados imediatamente da dependência.

Se você utilizar a opção **;pend** e o fluxo de job não tiver sido lançado, o cancelamento será deferido até que todas suas dependências, incluindo uma hora **at**, sejam resolvidas. Depois de todas as dependências terem sido resolvidas, o fluxo de job será cancelado e todos os jobs ou fluxos de job dependentes serão liberados da dependência. Durante o período em que **cancel** é deferido, a notação **[Cancel Pend]** é listada na coluna Dependências de uma tela **showschedules**.

Se você incluir a opção **;pend** e o fluxo de job já tiver sido lançado, os jobs restantes do fluxo de job serão cancelados e todos os jobs e fluxos de job dependentes serão liberados da dependência.

Para cancelar imediatamente um fluxo de job marcado com **[Cancel Pend]**, digite um comando **release** para o fluxo de job ou digite outro comando **cancel** sem a opção **;pend**.

Para parar a avaliação das dependências, configure a prioridade do fluxo de job como zero com o comando **altpri**. Para continuar a avaliá-las, defina a prioridade com um valor maior que zero.

Se o fluxo de tarefas cancelado contiver as tarefas definidas com a opção **every**, apenas a última instância dessas tarefas será listada como cancelada em uma exibição **showjobs**.

Exemplos

Para cancelar o fluxo de tarefas `sked1(1200 02/17/06)` na estação de trabalho `site2`, execute o seguinte comando:

```
cs site2#sked1(1200 02/17)
```

Para cancelar o fluxo de tarefas `mis2 (0900 02/19/06)` se estiver no estado STUCK, execute o seguinte comando:

```
cs mis2(0900 02/19)+state=stuck
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Monitorar Fluxos de Tarefas**.
2. Selecione **Todos os Fluxos de tarefas no plano** ou outra tarefa para monitorar fluxos de tarefas.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. Selecione um fluxo de tarefas e clique em **Mais Ações** > **Cancelar**.

checkhealthstatus

Chame o serviço **chkhltst** para verificar a conectividade entre o gerenciador de domínio e as estações de trabalho. Verifica se o arquivo Symphony não está corrompido, se os arquivos da caixa de correio podem ser lidos com sucesso pelo **mailman**, sem erros no cabeçalho do caixa de correio, e se a caixa de correio não está cheia. Esse comando pode ser útil para diagnosticar a razão para uma estação de trabalho desconectada e para obter sugestões sobre como recuperar o problema.

Sintaxe

{**checkhealthstatus** | **chs**} [*workstation*]

Comentários

Se *workstation* não for especificado, o serviço é ativado localmente.

Exemplos

Para verificar o status de funcionamento da estação de trabalho *site1*, ative o comando a seguir:

```
checkhealthstatus site1
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Ambiente** > **Monitorar Estações de Trabalho**.
2. Selecione uma tarefa para monitorar estações de trabalho.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. .

5. Na tabela que contém a lista de estações de trabalho, selecione a estação de trabalho cuja conectividade deseja verificar e clique na guia **Mais Ações** > **Verificar Status de Funcionamento...**

confirm

Confirma a conclusão de um job que foi planejado com a palavra-chave **confirmed**.

Você deve ter acesso ***confirm*** ao job.

Sintaxe

{**confirm** | **conf**} *jobselect*
;{**succ** | **abend**}
[**;noask**]

Argumentos

jobselect
Consulte "Selecionando Tarefas em Comandos" na página 345.

succ Confirma que o job terminou com êxito.

abend
Confirma que o job terminou sem êxito.

noask Especifica para não solicitar confirmação antes da tomada de ação em cada job qualificado.

Comentários

Alterar o estado de um job de ABEND para SUCC não requer que a palavra-chave **confirmed** seja utilizada para planejar o job. Para obter informações adicionais sobre confirmação da tarefa, consulte "confirmed" na página 222. Para obter informações adicionais sobre tarefas `EXTERNAL`, consulte "Gerenciando Dependências da Interligação de Redes no Plano" na página 620.

Tabela 66 mostra o efeito do comando **confirm** nos vários estados de tarefas:

Tabela 66. Alteração de Estado depois do Comando confirm

Estado Inicial do Job	Estado após confirm ;succ	Estado após confirm ;abend
READY	nenhum efeito	nenhum efeito
HOLD	nenhum efeito	nenhum efeito
EXEC	**SUCCP**	**ABENP**
ABENP	**SUCCP**	nenhum efeito
SUCCP	nenhum efeito	nenhum efeito
PEND	**SUCC**	**ABEND**
DONE	**SUCC**	**ABEND**
SUCC	nenhum efeito	nenhum efeito
ABEND	**SUCC**	nenhum efeito
FAIL	nenhum efeito	nenhum efeito
SCHED	nenhum efeito	nenhum efeito
ERROR (somente para tarefas sombra)	**SUCC**	**ABEND**
qualquer tarefa no fluxo de tarefas **EXTERNAL**	**SUCC**	**ABEND**

Exemplos

Para emitir uma confirmação **succ** para a tarefa `job3` no fluxo de tarefas `misdly(1200 02/17/06)`, execute o seguinte comando:

```
confirm misdly(1200 02/17/06).job3;succ
```

Para emitir uma confirmação **abend** para o job número **234**, execute o seguinte comando:

```
confirm 234;abend
```

console

Designa o console do Tivoli Workload Scheduler e configura o nível de mensagem.

Você deve ter acesso ***console*** à estação de trabalho.

Sintaxe

{**console** | **cons**}
[**sess** | **sys**]
[**;level=***msglevel*]

Argumentos

sess Envia mensagens e prompts do console do Tivoli Workload Scheduler para saída padrão.

sys Pára de enviar mensagens e prompts do console do Tivoli Workload Scheduler para saída padrão. Isso ocorre automaticamente quando você sai do **conman**.

msglevel
O nível das mensagens do Tivoli Workload Scheduler que são enviadas ao console. Especifique um dos seguintes níveis:

-1 Este é o valor que o produto automaticamente designa se você modificar qualquer dos argumentos para o console e nenhum valor for redesignado a `msglevel`. Com esse valor, o produto envia todas as mensagens geradas por todos os agentes e para todas as operações ao console.

0 Sem mensagens. Este é o padrão em agente tolerante a falhass.

1 Mensagens de exceção, tais como prompts do operador e abends de job.

2 Nível 1, mais as mensagens de êxito do fluxo de job.

3 Nível 2, mais as mensagens de êxito do job. Esse é o padrão no gerenciador de domínio principal.

4 Nível 3, mais as mensagens de jobs lançados.

Comentários

Se você digitar um comando **console** sem opções, o estado atual do console será exibido.

Por padrão, os processos de controle do Tivoli Workload Scheduler gravam mensagens e prompts do console em arquivos de lista padrão. No UNIX, você também pode enviá-los para o daemon**syslog**.

Exemplos

Para começar a gravar mensagens e avisos do console na saída padrão e alterar o nível de mensagem para **1**, execute o seguinte comando:

```
console sess;level=1
```

Para parar a gravação de mensagens do console e prompts na saída padrão e alterar o nível de mensagem para **4**, execute o seguinte comando:

```
cons sys;l=4
```

Para exibir o estado atual do console, execute o seguinte comando:

```
cons
Console is #J675, level 2, session
```

675 é o ID de processo da shell do usuário.

continue

Ignora o próximo erro de comando.

Sintaxe

{**continue** | **cont**}

Comentários

Esse comando é útil quando os comandos são digitados de forma não interativa. Ele orienta o **conman** a continuar a execução de comandos, mesmo que o próximo comando, depois de **continue**, resulte em erro.

Exemplos

Para que o **conman** continue com o comando **rerun** mesmo que o comando **cancel** falhe, execute o seguinte comando:

```
conman "cont&cancel=176&rerun job=sked5(1200 02/17/06).job3"
```

deldep job

Exclui dependências de um job.

Você deve ter acesso ***deldep*** ao job.

Sintaxe

{**deldep job** | **ddj**} *jobselect*
;*dependency*[**;**...]
[**;noask**]

Argumentos

jobselect
Consulte “Selecionando Tarefas em Comandos” na página 345.

dependência
O tipo de dependência. Especifique pelo menos uma das seguintes. É possível utilizar caracteres curingas em *workstation*, *jstream*, *job*, *resource*, *filename* e *promptname*.

at[**=***time* | *lowtime* | *hightime* | *lowtime*,*hightime*]

confirmed

deadline[=*time*[**timezone** | **tz** *tzname*][+*n* **days** | *mm/dd[/yy]*]]

every

follows=[*netagent*::][*workstation*#]{*jobstreamname*(*hhmm* [*mm/dd[/yy]*]) [*.job* | **@**] | *jobstream_id.job*;**schedid**} | *job*[**,**...]

maxdur=[*hhhmm*] [**onmaxdur** *action*]

mindur=[*hhhmm*] [**onmindur** *action*]

needs[=[*num*] [*workstation*#]*resource*[**,**...]]

opens[=[*workstation*#]"*filename*"[**(***qualifier***)**][**,**...]]

priority

prompt[="[**:** | **!**]*text*" | *promptname*[**,**...]]

until[=*time* [**timezone** | **tz** *tzname*][+*n* **day**[**s**]] [**;onuntil** *action*]]

noask Especifica para não solicitar confirmação antes da tomada de ação em cada job qualificado.

Comentários

Se você excluir `priority`, o job reverterá para sua prioridade planejada original. Ao excluir uma dependência `opens`, é possível incluir apenas o nome de arquivo base e o **conman** executará uma procura sem distinção de maiúsculas/minúsculas para correspondência de arquivos, ignorando os nomes de diretórios. As dependências de todos os arquivos correspondentes são excluídas.

As dependências excluídas não estão mais em efeito na execução do comando **rerun**.

Para excluir todas as dependências `follows` das tarefas contidas em um fluxo de tarefa específico, especifique a palavra-chave `follows` como:

```
follows=job_stream_name
```

Não utilize um curinga neste caso (tal como `follows=job_stream_name.@` porque o comando será rejeitado.

Exemplos

Para excluir uma dependência de recurso da tarefa `job3` no fluxo de tarefas `sked9(0900 02/19/06)`, execute o seguinte comando:

```
ddj sked9(0900 02/19/06).job3 ; needs=2 tapes
```

Para excluir todas as dependências follows externas do fluxo de tarefas `CPUA#TEST(0900 02/19/06)`, execute o seguinte comando:

```
ddj CPUA#TEST(0900 02/19/06).JOBA ; follows
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Monitorar Tarefas**.
2. Selecione **Todas as Tarefas no plano** ou outra tarefa para monitorar tarefas.

3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. Na tabela que contém a lista de tarefas, selecione a tarefa na qual você deseja incluir uma dependência e clique na guia **Dependências...**
5. No painel Dependências, expanda uma seção de dependência e clique no botão correspondente à ação você deseja executar.

deldep sched

Exclui dependências de um fluxo de job.

Você deve ter acesso ***deldep*** ao fluxo de job.

Sintaxe

{**deldep sched** | **dds**} *jstreamselect*
;*dependency*[**;**...]
[**;noask**]

Argumentos

jstreamselect
Consulte "Selecionando Tarefas em Comandos" na página 345.

dependência
O tipo de dependência. Especifique pelo menos uma das seguintes. É possível utilizar caracteres curinga em *workstation, jstreamname, jobname, resource, filename* e *promptname*.

at[**=***time* | *lowtime* | *hightime* | *lowtime*,*hightime*]

carryforward

deadline[**=***time*[**timezone** | **tz** *tzname*][**+***n* **days** | *mm/dd[/yy]*]]

follows=[*netagent***::**][*workstation***#**]{*jobstreamname*[*hhmm* [*mm/dd[/yy]*]][*.job* | **@**] | *jobstream_id.job***;schedid**}| *job*[**,**...]

limite

needs[**=**[*num*] [*workstation***#**]*resource*[**,**...]]

opens[**=**[*workstation***#**]"*filename*"[**(***qualifier***)**][**,**...]]

priority

prompt[**="[:** | **!]***text***"** | *promptname*[**,**...]]

until[**=***time* [**timezone**|**tz** *tzname*][**+***n* **day**[**s**]] [**;onuntil** *action*]]

noask Especifica para não solicitar confirmação antes da tomada de ação em cada fluxo de job qualificado.

Comentários

Se você excluir `priority`, o job reverterá para sua prioridade planejada original. Ao excluir uma dependência `opens`, é possível incluir apenas o nome de arquivo base e o **conman** executará uma procura sem distinção de maiúsculas/minúsculas para correspondência de arquivos, ignorando os nomes de diretórios. As dependências de todos os arquivos correspondentes são excluídas.

As dependências excluídas não estão mais em efeito na execução do comando **rerun**.

Exemplos

Para excluir uma dependência `resource` do fluxo de tarefas `sked5(0900 02/19/06)`, execute o seguinte comando:

```
dds sked5(0900 02/19/06);needs=2 tapes
```

Para excluir todas as dependências `follows` do fluxo de tarefas `sked3(1000 04/19/06)`, execute o seguinte comando:

```
dds sked3(1000 04/19/06);follows
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Monitorar Fluxos de Tarefas**.
2. Selecione **Todos os Fluxos de tarefas no plano** ou outra tarefa para monitorar fluxos de tarefas.
3. Escolha um nome de mecanismo ou especifique as propriedades da conexão e clique em **OK**
4. Na tabela que contém a lista de fluxos de tarefas, selecione o fluxo de tarefas no qual deseja incluir uma dependência e clique na guia **Dependências...**
5. No painel Dependências, expanda uma seção de dependência e clique no botão correspondente à ação você deseja executar.

deployconf

Obtém a configuração de monitoramento mais recente para o mecanismo de monitoramento de eventos na estação de trabalho.

Sintaxe

{**deployconf** | **deploy**} [*domain***!**]*workstation*

Argumentos

domínio

Especifica o nome do domínio de destino para a operação. Caracteres curinga são permitidos.

Este argumento é útil ao implementar mais de uma estação de trabalho em um domínio. Por exemplo, para implementar a configuração de monitoramento mais recente para todos os agentes no domínio `AURORABU`, utilize o seguinte comando:

```
deploy
AURORABU!@
```

O domínio não é necessário se você não incluir caracteres curingas em *workstation*.

Se você não incluir *domain* e incluir caracteres curinga em *workstation*, o domínio padrão será aquele no qual o **conman** está sendo executado.

estação de trabalho

Especifica o nome da estação de trabalho na qual o mecanismo de monitoramento é executado. Caracteres curinga não são permitidos.

Comentários

Se a configuração existente já estiver atualizada, o comando não terá efeito.

A permissão para ações `start` nos objetos `cpu` deve ser ativada no arquivo de segurança para executar este comando.

display

Exibe um arquivo de job ou uma definição de fluxo de job.

Se você especificar um arquivo pelo nome, deverá ter acesso read ao arquivo. Para arquivos de job e definições de fluxo de job, você deve ter acesso ***display*** ao job ou fluxo de job.

Sintaxe

{**display file** | **df**} *filename* [**;offline**]

{**display job** | **dj**} *jobselect* [**;offline**]

{**display sched** | **ds**} *jstreamselect*
[**valid** {**at** *date* | **in** *date date*}
[**;offline**]

Argumentos

nome do arquivo
Especifica o nome do arquivo, geralmente, um arquivo de script de job. O nome deverá ser colocado entre aspas (") se contiver caracteres diferentes dos seguintes: alfanuméricos, traços (-), barras (/), barras invertidas (\) e sublinhados (_). Caracteres curinga são permitidos. O arquivo deve ser acessível da estação de trabalho de logon. Utilize essa opção se quiser mostrar somente o conteúdo do arquivo de script da tarefa.

jobselect
O job cujo arquivo de job é exibido. Consulte "Selecionando Tarefas em Comandos" na página 345. O arquivo de job deve ser acessível da estação de trabalho de logon. Essa palavra-chave aplica-se somente ao caminho e ao nome de arquivo do arquivo de script de tarefas definidas com a opção **scriptname**.

jstreamselect
O fluxo de job cuja definição é exibida. Consulte "Selecionando Fluxos de Tarefas em Comandos" na página 353.

valid Especifica o dia ou o intervalo de dias durante os quais as instâncias do fluxo de tarefa a serem exibidas devem estar ativas. Isso significa que o intervalo de validade dessas instâncias do fluxo de tarefa deve conter o espaço de tempo especificado no argumento **valid**. O formato utilizado para *date* depende do valor designado para a variável ***date format*** especificada no arquivo `localopts`. Se não for especificado, a instância selecionada é a única válida hoje.

offline
Envia a saída do comando para o dispositivo de saída do **conman**. Para obter informações sobre esse dispositivo, consulte "Saída Off-line" na página 340.

Exemplos

Para exibir o arquivo `c:\maestro\jclfiles\arjob3`, execute o seguinte comando:

```
df c:\apps\maestro\jclfiles\arjob3
```

Para exibir o arquivo de script para a tarefa `createpostreports` no fluxo de tarefas `FINALPOSTREPORTS`, execute o seguinte comando:

```
dj FINALPOSTREPORTS(2359 02/14/13).CREATEPOSTREPORTS
```

Esta é uma amostra de saída desse comando:

```
M235062_99#FINALPOSTREPORTS(2359 02/14/13).CREATEPOSTREPORTS /opt/TWA/TWS/
                                                              CreatePostReports
   #!/bin/sh
  ##################################################################
  # Materiais Licenciados - Propriedade da IBM
  # Materiais Restritos da IBM
  # 5698-WSH
  # (C) Copyright IBM Corp. 1998, 2011 Todos os Direitos Reservados.
  # Direitos Restritos para Usuários do Governo dos Estados Unidos - Uso, duplicação ou
  # divulgação restritos pelo documento GSA ADP Schedule Contract com a IBM Corp.
  ##################################################################
  #@(#) $Id: CreatePostReports.sh,v 1.0

  ##
  ## Definições do catálogo de mensagens CreatePostReports.
  ##

  ##
  ## ID do conjunto de mensagens
  ##
  MAE_CREATEPOSTREPORTS_SET=226
  MAE_COPYRIGHT_SET=234

  ##
  ....
  ...
  ....
  #
  # Fim
  #
```

Para exibir a definição de fluxo de tarefas para o fluxo de tarefas `mod`, execute o seguinte comando:

```
ds mod
```

Esta é uma amostra de saída desse comando:

```
Job Stream Name Workstation  Valid From Updated On Locked By
 ---------------  ----------------  ----------  ----------  ----------------
MOD    M235062_99  06/30/2007 03/04/2006 -

SCHEDULE M235062_99#MOD VALIDFROM 06/30/2007
ON RUNCYCLE SCHED1_PREDSIMPLE VALIDFROM 07/18/2007
"FREQ=DAILY;INTERVAL=1"
( AT 1111 )
CARRYFORWARD
FOLLOWS M235062_99#SCHED_FIRST1.@
FOLLOWS M235062_99#SCHED_FIRST.JOB_FTA
PRIORITY 66
:
M235062_99#JOBMDM
 SCRIPTNAME "/usr/acct/scripts/gl1" STREAMLOGON root
 DESCRIPTION "general ledger job1"
 TASKTYPE UNIX
 RECOVERY STOP
 PRIORITY 30
NEEDS 16 M235062_99#JOBSLOTS
PROMPT PRMT3

B236153_00#JOB_FTA
FOLLOWS M235062_99#SCHED_FIRST1.@
FOLLOWS M235062_99#SCHED_FIRST.JOB_FTA
PRIORITY 66
:
```

```
M235062_99#JOBMDM
 SCRIPTNAME "/usr/acct/scripts/gl1" STREAMLOGON root
 DESCRIPTION "general ledger job1"
 TASKTYPE UNIX
 RECOVERY STOP
 PRIORITY 30
NEEDS 16 M235062_99#JOBSLOTS
PROMPT PRMT3

B236153_00#JOB_FTA
 DOCOMMAND "echo pippo" STREAMLOGON root
 DESCRIPTION "general ledger job1"
 TASKTYPE UNIX
 RECOVERY STOP
 FOLLOWS JOBMDM
END
```

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

o Guia do Usuário do Console do Dynamic Workload, seção Listando Definições de Objetos no Banco de Dados.

Para obter informações adicionais sobre como criar e editar objetos de planejamento, consulte:

o Guia do Usuário do Console do Dynamic Workload, seção Projetando sua Carga de Trabalho.

exit

Sai do programa de linha de comandos do **conman**.

Sintaxe

{**exit** | **e**}

Comentários

Quando você está no modo de ajuda no UNIX, esse comando retorna o **conman** para o modo de entrada de comandos.

Exemplos

Para sair do programa de linha de comandos **conman**, execute o seguinte comando:

```
exit
```

ou

```
J
```

fence

Altera o limitador de job em uma estação de trabalho. Os jobs não serão lançados na estação de trabalho se suas prioridades forem menores ou iguais ao limitador de job.

Você deve ter acesso *fence* à estação de trabalho.

Sintaxe

{**fence** | **f**} *workstation*
;pri
[**;noask**]

Argumentos

estação de trabalho
Especifica o nome da estação de trabalho. O padrão é sua estação de trabalho de logon.

pri Especifica o nível de prioridade. É possível digitar de **0** a **99**, **hi**, **go** ou **system**. Digitar **system** define o limitador de job como zero.

noask Especifica para não solicitar confirmação antes da tomada de ação em cada estação de trabalho qualificada.

Comentários

O limitador de job evita o lançamento de jobs de baixa prioridade, independente das prioridades de seus fluxos de job. É possível, portanto, manter jobs de baixa prioridade em alta prioridade, enquanto se permite que jobs de alta prioridade em fluxos de job de baixa prioridade sejam lançados.

Quando você inicia o Tivoli Workload Scheduler pela primeira vez após a instalação, o limitador de job é definido como zero. Ao alterar a fence da tarefa, é transportado durante o processamento de pré-produção no plano de produção do próximo dia.

Para exibir a definição atual do limitador de job, utilize o comando **status**.

Exemplos

Para alterar a cerca de job na estação de trabalho `site4`, execute o seguinte comando:

```
fence site4;20
```

Para alterar a cerca de job na estação de trabalho na qual o **conman** está sendo executado, execute o seguinte comando:

```
f ;40
```

Para impedir que todas as tarefas sejam ativadas pelo Tivoli Workload Scheduler na estação de trabalho `tx3`, execute o seguinte comando:

```
f tx3;go
```

Para alterar a cerca de job para zero na estação de trabalho na qual o **conman** está sendo executado, execute o seguinte comando:

```
f ;system
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Monitorar Tarefas** ou **Monitorar Fluxos de Tarefas**.

2. Selecione uma tarefa de monitor.
3. Escolha um nome de mecanismo ou especifique as propriedades da conexão e clique em **OK**
4. Na tabela que contém a lista de objetos, selecione a tarefa ou fluxo de tarefas cuja prioridade você deseja alterar e clique na guia **Mais Ações** > **Fence...**
5. No painel exibido, configure a nova prioridade e clique em **OK**.

help

Exibe informações de auxílio sobre comandos. Não disponível no Windows.

Sintaxe

{**help** | **h**} {*command* | *keyword*}

Argumentos

comando

Especifica o nome de um comando do **conman** ou do sistema. Para comandos do **conman**, digite o nome completo do comando, pois abreviações e formas simplificadas não são suportadas. Para comandos que consistem em duas palavras, digite a primeira palavra e será exibida ajuda para todas as versões do comando. Por exemplo, digitar **help display** exibe informações sobre os comandos **display file**, **display job** e **display sched**.

keyword

Você também pode digitar as seguintes palavras-chave:

COMMANDS
Lista todos os comandos do conman.

SETUPCONMAN
Descreve como configurar para usar o conman.

RUNCONMAN
Como executar conman.

SPECIALCHAR
Descreve curingas, delimitadores e outros caracteres especiais que é possível usar.

JOBSELECT
Lista informações sobre a seleção de jobs para comandos.

JOBSTATES
Lista informações sobre os estados do job.

JSSELECT
Lista informações sobre a seleção de fluxos de job para comandos.

JSSTATES
Lista informações sobre os estados do fluxo de job.

MANAGEBACKLEVEL
Gerenciando tarefas e fluxos de tarefas a partir de agentes de nível anterior.

Exemplos

Para exibir uma lista de todos os comandos do **conman**, execute o seguinte comando:

```
help commands
```

Para exibir informações sobre o comando **fence**, execute o seguinte comando:

```
help fence
```

Para exibir informações sobre os comandos **altpri job** e **altpri sched**, execute o seguinte comando:

```
h altpri
```

Para exibir informações sobre caracteres especiais que é possível usar, execute o comando a seguir:

```
h specialchar
```

kill

Pára um job que está em execução. No UNIX, isso é realizado com um comando UNIX **kill**. Você deve ter acesso ***kill*** ao job.

Sintaxe

{**kill** | **k**} *jobselect*
[;noask]

Argumentos

jobselect
Consulte "Selecionando Tarefas em Comandos" na página 345.

noask Especifica para não solicitar confirmação antes da tomada de ação em cada job qualificado.

Comentários

A operação **kill** não é executada pelo **conman** e sim por um processo de produção do Tivoli Worload Scheduler, portanto, pode haver um pequeno retardo.

Os jobs eliminados terminam no estado ABEND. Todos os jobs ou fluxos de job dependentes de um job eliminado não são liberados. Jobs eliminados podem ser executados novamente.

Exemplos

Para eliminar a tarefa `report` no fluxo de tarefas `apwkly(0600 03/05/06)` na estação de trabalho `site3`, execute o seguinte comando:

```
kill site3#apwkly(0600 03/05/06).report
```

Para efetuar kill da tarefa número `124` em execução na estação de trabalho `geneva`, execute o comando a seguir:

```
kill geneva#124
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Monitorar Tarefas**.
2. Selecione **Todas as Tarefas no plano** ou outra tarefa para monitorar tarefas.

3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. Na tabela que contém a lista de tarefas, selecione a tarefa na qual você deseja incluir uma dependência e clique na guia **Dependências...**
5. No painel Dependências, expanda uma seção de dependência e clique no botão correspondente à ação você deseja executar.

limit cpu

Altera o limite de jobs que podem ser executados simultaneamente em uma estação de trabalho. Você deve ter acesso ***limit*** à estação de trabalho.

Sintaxe

{**limit cpu** | **lc** } *workstation*
;*limit*
[**;noask**]

Argumentos

estação de trabalho
Especifica o nome da estação de trabalho. Caracteres curinga são permitidos. O padrão é sua estação de trabalho de logon.

limite Especifica quantas tarefas podem ser executadas simultaneamente na estação de trabalho. Valores suportados vão de **0** a **1024** ed **system**.

Se você configurar limit cpu para **0**:
- Para um fluxo de tarefas no estado **READY**, apenas as tarefas com valores de prioridade **hi** e **go** podem ser ativadas na estação de trabalho.
- Para um fluxo de tarefas com um valor de prioridade **hi** ou **go**, todas as tarefas com um valor de prioridade diferente de 0 podem ser ativadas na estação de trabalho.

Se você configurar limit cpu para **system**, não há nenhum limite para o número de tarefas simultâneas na estação de trabalho.

noask Especifica para não solicitar confirmação antes da tomada de ação em cada estação de trabalho qualificada.

Comentários

Para exibir o limite atual do job na estação de trabalho de logon, utilize o comando **status**.

Quando você inicia o Tivoli Workload Scheduler pela primeira vez após a instalação, o limite de job da estação de trabalho é definido como zero e deve ser elevado antes da ativação dos jobs. Ao alterar o limite, ele é transportado durante o processamento de pré-produção no plano de produção do próximo dia.

O Tivoli Workload Scheduler tenta lançar o máximo de tarefas possível dentro do limite de tarefas. Há um limite prático para o número de processos que podem ser iniciados em uma estação de trabalho. Se esse limite for alcançado, o sistema responderá com uma mensagem indicando que os recursos do sistema não estão disponíveis. Quando um job não pode ser lançado por este motivo, ele entra no estado **fail**. Abaixar o limite do job pode evitar que isso ocorra.

Exemplos

Para alterar o limite de job na estação de trabalho na qual o **conman** está sendo executado, execute o seguinte comando:

```
lc ;12
```

Para alterar o limite de job na estação de trabalho `rx12`, execute o seguinte comando:

```
lc rx12;6
```

Para configurar como 10 o limite de tarefas em todas as estações de trabalho pertencentes ao domínio e aos domínios filhos, execute o seguinte comando:

```
lc @!@;10
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Ambiente** > **Monitorar Estações de Trabalho**.
2. Selecione uma tarefa para monitorar estações de trabalho.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. Na tabela contendo a lista de estações de trabalho, selecione uma estação de trabalho e clique em **Mais Ações** > **Limitar...**

limit sched

Altera o **limit** definido na definição de um fluxo de job. Para obter informações adicionais sobre como configurar um limite em uma definição de fluxo de tarefas, consulte “limite” na página 238. Você deve ter acesso ***limit*** ao fluxo de tarefa.

Sintaxe

{**limit sched** | **ls** } *jstreamselect*
;*limit*
[**;noask**]

Argumentos

jstreamselect
Consulte “Selecionando Fluxos de Tarefas em Comandos” na página 353.

limite Especifica o limite do job. Você pode digitar de **0** a **1024**.

noask Especifica para não solicitar confirmação antes da tomada de ação em cada fluxo de job qualificado.

Exemplos

Para alterar o limite de job em todos os fluxos de job que incluem `sales` no nome, execute o seguinte comando:

```
ls sales@;4
```

Para alterar o limite de tarefa no fluxo de tarefas `CPUA#Job1`, execute o seguinte comando:

```
ls CPUA#apwkly;6
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Monitorar Fluxos de Tarefas**.
2. Selecione **Todos os Fluxos de tarefas no plano** ou outra tarefa para monitorar fluxos de tarefas.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. Selecione um fluxo de tarefas e clique em **Mais Ações** > **Limitar...**

link

Abre links de comunicação entre estações de trabalho. Em uma rede do Tivoli Workload Scheduler, os agentes tolerantes a falhas e padrão são ligados a seus gerenciadores de domínio e os gerenciadores de domínio são ligados a seus gerenciadores de domínio pai. Agentes estendidos não são ligados; sua comunicação é feita através de um host.

Você deve ter acesso ***link*** à estação de trabalho de destino.

Sintaxe

{**link** | **lk**} [*domain***!**]*workstation*
[**;noask**]

Argumentos

domínio
Especifica o nome do domínio no qual os links são abertos. Caracteres curinga são permitidos.

Este argumento é útil quando se está ligando mais de uma estação de trabalho em um domínio. Por exemplo, para ligar todos os agentes no domínio `stlouis`, utilize o seguinte comando:

```
lk stlouis!@
```

O domínio não é necessário se você não incluir caracteres curingas em *workstation*.

Se você não incluir *domain* e incluir caracteres curinga em *workstation*, o domínio padrão será aquele no qual o **conman** está sendo executado.

estação de trabalho
Especifica o nome da estação de trabalho a ser ligada. Caracteres curinga são permitidos.

Esse comando não é suportado em estações de trabalho de mecanismo remoto.

noask Especifica para não solicitar confirmação antes da tomada de ação em cada estação de trabalho qualificada.

Comentários

Se a opção **autolink** for definida como **on** em uma definição de estação de trabalho, seu link será aberto automaticamente sempre que o Tivoli Workload Scheduler for iniciado. Se **autolink** for definido como **off**, você deverá utilizar os

comandos **link** e **unlink** para controlar a ligação. Para obter informações sobre **autolink**, consulte “Definição de estação de trabalho” na página 139.

Supondo-se que um usuário tenha o acesso **link** a estações de trabalho sendo ligadas, as seguintes regras serão aplicadas:

- Um usuário executando o **conman** no gerenciador de domínio principal pode vincular qualquer estação de trabalho na rede.
- Um usuário executando o **conman** em um gerenciador de domínio diferente do principal pode vincular qualquer estação de trabalho em seu próprio domínio e em domínios subordinados. O usuário não pode ligar estações de trabalho em domínios ponto a ponto.
- Um usuário executando o **conman** em um agente pode vincular qualquer estação de trabalho em seu domínio local, contanto que a estação de trabalho seja um gerenciador de domínio ou um host. Um agente de período no domínio local não pode ser vinculado.
- Para vincular um domínio subordinado durante a execução do **conman** em um domínio superior, não é necessário que os links intervenientes estejam abertos.

Exemplos

A Figura 23 e a Tabela 67 na página 388 mostram os links abertos pelos comandos **link** executados pelos usuários em vários locais da rede.
DM*n* são gerenciadores de domínio e **A***nn* são agentes.

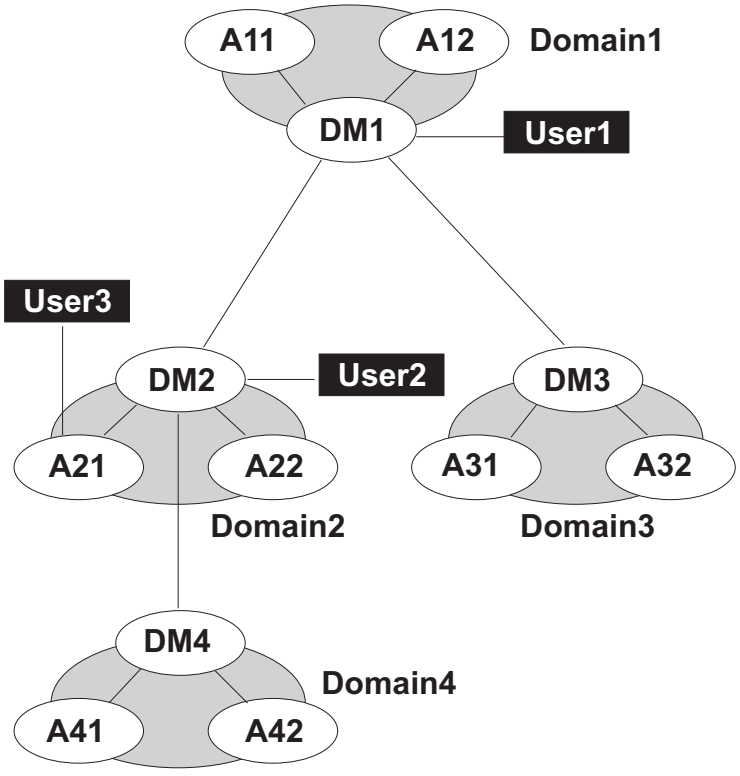


Figura 23. Links da Rede

Tabela 67. Links Abertos

Comando	Links Abertos pelo Usuário1	Links Abertos pelo Usuário2	Links Abertos pelo Usuário3
link @!@	Todos os links são abertos.	DM1-DM2 DM2-A21 DM2-A22 DM2-DM4 DM4-A41 DM4-A42	DM2-A21
link @	DM1-A11 DM1-A12 DM1-DM2 DM1-DM3	DM1-DM2 DM2-A21 DM2-A22 DM2-DM4	DM2-A21
link DOMAIN3!@	DM3-A31 DM3-A32	Não permitido.	Não permitido.
link DOMAIN4!@	DM4-A41 DM4-A42	DM4-A41 DM4-A42	Não permitido.
link DM2	DM1-DM2	Não aplicável.	DM2-A21
link A42	DM4-A42	DM4-A42	Não permitido.
link A31	DM3-A31	Não permitido.	Não permitido.

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Ambiente** > **Monitorar Estações de Trabalho**.
2. Selecione uma tarefa para monitorar estações de trabalho.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. Na tabela contendo a lista de estações de trabalho, selecione uma estação de trabalho e clique em **Mais Ações** > **Vincular...**

listsym

Lista o plano de produção (arquivos Symphony) já processado.

Sintaxe

{**listsym** | **lis**} [**trial** | **forecast**]
[**;offline**]

Argumentos

trial Lista planos trial.

previsão
Lista planos de previsão.

off-line
Envia a saída do comando para o dispositivo de saída do **conman**. Para obter informações sobre esse dispositivo, consulte "Saída Off-line" na página 340.

Resultados

Programar Data
A data utilizada para selecionar os fluxos de tarefa a serem executados.

Data Real
A data em que **batchman** começou a executar o arquivo Symphony.

Horário de Início
A hora em que **batchman** começou a executar o arquivo Symphony.

Data de Registro
A data em que o plano (arquivo Symphony) foi registrado pelo comando **stageman**.

Núm. de Execução
O número de execução designado ao plano (arquivo Symphony). É utilizado internamente para sincronização da rede do Tivoli Workload Scheduler.

Tamanho
O número de registros contidos no arquivo Symphony.

Núm. de Log
O número de log que indica a ordem cronológica dos arquivos de log. Esse número pode ser utilizado em um comando **setsym** para mudar para um arquivo de log específico.

Nome do Arquivo
O nome do arquivo de log atribuído pelo comando **stageman**.

Exemplos

Para listar os arquivos do plano de produção, execute o seguinte comando:

```
listsym
```

esta é uma saída de amostra para o comando:

```
Job Stream Actual Start  Log  Run   Log
Date   Date Time  Date  Num Size Num Filename
03/05/06 03/05/06 21:06 03/05/06  42 534  1 M200603052111 Exp
03/04/06 03/04/06 15:59 03/05/06  41 463  2 M200603052106 Exp
03/04/06 03/04/06 15:51 03/04/06  40 362  3 M200603041559 Exp
03/04/06 03/04/06 14:31 03/04/06  39 460  4 M200603041551 Exp
03/04/06 03/04/06 14:26 03/04/06  38 436  5 M200603041431 Exp
03/04/06 03/04/06 14:24 03/04/06  37 436  6 M200603041426 Exp
03/04/06 03/04/06 14:19 03/04/06  36 436  7 M200603041424 Exp
03/04/06 03/04/06 14:17 03/04/06  35 436  8 M200603041419 Exp
03/04/06 03/04/06 14:17 03/04/06  34 364  9 M200603041417 Exp
```

Para listar arquivos que contêm planos trial, execute o seguinte comando:

```
listsym trial
```

esta é uma saída de amostra para o comando:

```
Job Stream Actual Start  Log  Run   Log
Date   Date Time  Date  Num Size Num Filename
03/03/06     03/03/06  0 126  1 Tpippo   Exp
03/03/06     03/03/06  0 1850  2 Tangelo2   Exp
03/03/06     03/03/06  0 1838  3 Tangelo1   Exp
```

Para listar os arquivos que contém os planos de previsão, execute o seguinte comando:

```
listsym forecast
```

Esta é uma saída de amostra para o comando:

```
Job Stream Actual Start  Log  Run   Log
Date   Date Time  Date  Num Size Num Filename
03/03/06     03/03/06  0  62  1 Fpluto   Exp
```

Consulte também

No Dynamic Workload Console:

1. Na barra de navegação, clique em **Planejando** > **Previsão de Carga de Trabalho** > **Gerenciar Planos Disponíveis**.

2. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
3. Clique em um tipo de plano ou grave um nome do arquivo de plano
4. Clique em **Exibir Lista de Planos**.

recall

Exibe prompts que estão aguardando uma resposta.

Você deve ter acesso ***display*** aos prompts.

Sintaxe

{**recall** | **rc**} [*workstation*]
[**;offline**]

Argumentos

estação de trabalho
Especifica o nome da estação de trabalho na qual o prompt foi emitido. Se você não especificar uma estação de trabalho, somente os prompts da estação de trabalho de logon e os prompts globais serão exibidos.

off-line
Envia a saída do comando para o dispositivo de saída do **conman**. Para obter informações sobre esse dispositivo, consulte "Saída Off-line" na página 340.

Resultados

Estado
O estado do prompt. O estado de avisos pendentes é sempre ASKED.

Mensagem ou Prompt
Para prompts nomeados, o número da mensagem, o nome do prompt e o texto da mensagem. Para prompts não nomeados, o número da mensagem, o nome do job ou fluxo de job e o texto da mensagem.

Exemplos

Para exibir prompts pendentes emitidos na estação de trabalho na qual o **conman** está sendo executado, execute o seguinte comando:

```
recall
```

ou:

```
rc
```

Para exibir prompts pendentes na estação de trabalho`site3`, execute o seguinte comando:

```
rc site3
```

Para exibir prompts pendentes em todas as estações de trabalho e fazer com que a saída seja enviada para o dispositivo offline de **conman**, execute o comando a seguir:

```
rc @;off-line
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Prompts do Monitor**.
2. Selecione **Todos os Prompts no plano**, que listará todos os prompts, independentemente de seu status, ou criará e selecionará outra tarefa predefinida que lista apenas prompts no status ASKED.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.

redo

Edita e executa o comando anterior novamente.

Sintaxe

{**redo** | **red**}

Contexto

Quando você executa o comando **redo**, o **conman** exibe o comando anterior para que possa ser editado e executado novamente. Utilize a barra de espaços para mover o cursor sob o caractere a ser modificado e digite as diretrizes a seguir.

Diretrizes

d[*dir***]** Exclui o caractere acima de **d**. Pode vir seguido por outras diretrizes.

i*text* Insere texto antes do caractere acima de **i**.

r*text* Substitui um ou mais caracteres por texto, começando com o caractere acima de **r**. A substituição estará implícita se nenhuma outra diretriz for digitada.

>*text* Anexa texto no final da linha.

>d[*dir* | *text***]**
Exclui caracteres no final da linha. Pode vir seguido por outra diretriz ou texto.

>r*text* Substitui caracteres no final da linha por texto.

Exemplos de Diretrizes

ddd Exclui os três caracteres acima de **d**s.

iabc Insere **abc** antes do caractere acima de **i**.

rabc Substitui os três caracteres, começando pelo caractere acima de **r**, por **abc**.

abc Substitui os três caracteres acima de **abc** por **abc**.

d diabc
Exclui o caractere acima do primeiro**d**, pula um caractere, exclui o caractere acima do segundo**d** e insere **abc** em seu lugar.

>abc Anexa **abc** no final da linha.

>ddabc
Exclui os dois últimos caracteres na linha e insere **abc** em seu lugar.

>rabc Substitui os três últimos caracteres na linha por **abc**.

Exemplos

Para inserir um caractere, execute o seguinte comando:

```
redo
setsm 4
    iy
setsym 4
```

Para substituir um caractere, execute o seguinte comando:

```
redo
setsym 4
       5
setsym 5
```

release job

Libera jobs das dependências.

Você deve ter acesso ***release*** ao job.

Sintaxe

{**release job** | **rj**} *jobselect*
[**;***dependency*[**;**...]]
[**;noask**]

Argumentos

jobselect
Especifica o job ou jobs a serem liberados. Consulte "Selecionando Tarefas em Comandos" na página 345.

dependência
O tipo de dependência. É possível especificar uma das seguintes. É possível utilizar caracteres curinga em *workstation*, *jstreamname*, *jobname*, *resource*, *filename* e *promptname*.

at[**=***time* | *lowtime* | *hightime* | *lowtime*,*hightime*]

confirmed

deadline[**=***time*[**timezone** | **tz** *tzname*][**+***n* **days** | *mm/dd[/yy]*]]

every

follows=[*netagent***::**][*workstation***#**]{*jobstreamname*[*hhmm* [*mm/dd[/yy]*]][*.job* | **@**] | *jobstream_id.job***;schedid**} | *job*[**,**...]

needs[**=**[*num*] [*workstation***#**]*resource*[**,**...]]

opens[**=**[*workstation***#**]"*filename*"[**(***qualifier***)**][**,**...]]

priority

prompt[**="[:** | **!**]*text***"** | *promptname*[**,**...]]

until[**=***time* [**timezone**|**tz** *tzname*][**+***n* **day**[**s**]] [**;onuntil** *action*]]

noask Especifica para não solicitar confirmação antes da tomada de ação em cada job qualificado.

Comentários

Ao liberar uma dependência `opens`, é possível incluir apenas o nome de arquivo base e o**conman** executará uma procura sem distinção de maiúsculas/minúsculas para correspondência de arquivos, ignorando os nomes de diretórios. As dependências de todos os arquivos correspondentes são liberadas.

Para dependências `needs`, o job liberado recebe o número requerido de unidades do recurso, apesar de poderem não estar disponíveis. Isso pode fazer com que as unidades disponíveis em um **showresources** exibam um número negativo.

Quando você libera um job de uma dependência `priority`, o job é revertido para sua prioridade planejada original.

As dependências liberadas permanecem em efeito ao executar o comando **rerun**.

Exemplos

Para liberar a tarefa `job3` no fluxo de tarefas `ap(1000 03/05/06)` de todas suas dependências, execute o seguinte comando:

```
rj ap(1000 03/05/06).job3
```

Para liberar todos os jobs na estação de trabalho`site4` de suas dependências em um aviso chamado`glprmt`, execute o seguinte comando:

```
rj site4#@.@;prompt=glprmt
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Monitorar Tarefas**.
2. Selecione **Todas as Tarefas no plano** ou outra tarefa para monitorar tarefas.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. Na tabela que contém a lista de tarefas, selecione uma tarefa e clique em **Mais Ações** > **Liberar**.

release sched

Libera fluxos de job das dependências.

Você deve ter acesso ***release*** ao fluxo de job.

Sintaxe

{**release sched** | **rs**} *jstreamselect*
[**;***dependency*[**;**...]]
[**;noask**]

Argumentos

jstreamselect
Consulte “Selecionando Fluxos de Tarefas em Comandos” na página 353.

dependência

O tipo de dependência. Especifique uma das seguintes. É possível utilizar caracteres curinga em *workstation, jstream, job, resource, filename* e *promptname*.

at[**=***time* | *lowtime* | *hightime* | *lowtime*,*hightime*]

carryforward

deadline[**=***time*[**timezone** | **tz** *tzname*][**+***n* **days** | *mm/dd[/yy]*]]

follows=[*netagent***::**][*workstation***#**]{*jobstreamname*[*hhmm* [*mm/dd[/yy]*]][*.job* | **@**] | *jobstream_id.job***;schedid**} | *job*[**,**...]

limite

needs[**=**[*num*] [*workstation***#**]*resource*[**,**...]]

opens[**=**[*workstation***#**]"*filename*"[**(***qualifier***)**][**,**...]]

priority

prompt[**="[:** | **!**]*text***"** | *promptname*[**,**...]]

until[**=***time* [**timezone**|**tz** *tzname*][**+***n* **day**[**s**]] [**;onuntil** *action*]]

noask Especifica para não solicitar confirmação antes da tomada de ação em cada fluxo de job qualificado.

Comentários

Ao excluir uma dependência `opens`, é possível incluir apenas o nome de arquivo base e o**conman** executará uma procura sem distinção de maiúsculas/minúsculas para correspondência de arquivos, ignorando os nomes de diretórios. As dependências de todos os arquivos correspondentes são liberadas.

Para dependências `needs`, o fluxo de job liberado recebe o número requerido de unidades do recurso, apesar de poderem não estar disponíveis. Isso pode fazer com que as unidades disponíveis em um **showresources** exibam um número negativo.

Ao liberar um fluxo de tarefas uma dependência `priority`, o fluxo de tarefas é revertido para sua prioridade original.

Em determinadas circunstâncias, quando você submete um comando **deldep**, ele pode ter sido bem-sucedido, embora seja novamente redirecionado para o **batchman**. Para obter informações adicionais, consulte “Processamento de comandos do conman” na página 344.

Exemplos

Para liberar a instância do fluxo de tarefas com *jobstream_id* `0AAAAAAAAAAAABSE` de todas as suas dependências, execute o seguinte comando:

```
rs 0AAAAAAAAAAAABSE; schedid
```

Para liberar o fluxo de tarefas `sked5(1105 03/07/06)` de todas suas dependências `opens`, execute o seguinte comando:

```
rs sked5(1105 03/07/06);opens
```

Para liberar todos os fluxos de job na estação de trabalho`site3` de suas dependências no fluxo de job `main#sked23`, execute o seguinte comando:

```
rs site3#0;follows=main#sked23
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Monitorar Fluxos de Tarefas**.
2. Selecione **Todos os Fluxos de tarefas no plano** ou outra tarefa para monitorar fluxos de tarefas.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. Selecione um fluxo de tarefas e clique em **Mais Ações** > **Liberar**.

reply

Responde a um prompt de job ou fluxo de job.

Você deve ter acesso ***reply*** ao prompt nomeado ou global. Para responder a um prompt não nomeado, você deve ter acesso ***reply*** aos prompts e acesso ***reply*** ao job ou fluxo de job associado.

Sintaxe

{**reply** | **rep**}
{ *promptname* | [*workstation***#**]*msgnum*}
;*responder*
[**;noask**]

Argumentos

promptname
Especifica o nome de um prompt global. Caracteres curinga são permitidos.

estação de trabalho
Especifica o nome da estação de trabalho na qual um prompt não nomeado foi emitido.

msgnum
Especifica o número de mensagem de um prompt não nomeado. É possível exibir números de mensagens com os comandos **recall** e **showprompts**.

reply Especifica a resposta, **Y** para sim ou **N** para não.

noask Especifica para não solicitar confirmação antes da tomada de ação em cada prompt qualificado.

Comentários

Se a resposta for **Y**, as dependências no prompt serão satisfeitas. Se a resposta for **N**, as dependências não são atendidas e o prompt não é emitido novamente.

Os prompts podem ser respondidos antes de serem emitidos. É possível utilizar o comando **showprompts** para exibir todos os prompts, se tiverem ou não sido emitidos.

Exemplos

Para responder **S** ao aviso global `arpmt`, execute o seguinte comando:

```
reply arprmt;y
```

Para responder **N** ao número da mensagem **24** na estação de trabalho`site4`, execute o seguinte comando:

```
rep site4#24;n
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Prompts do Monitor**.
2. Selecione **Todos os Prompts no Plano** ou outra tarefa para monitorar prompts.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. Na tabela de resultados, selecione um prompt e clique em **Responder Sim** ou em **Responder Não**.

rerun

executa novamente uma tarefa.

Você deve ter acesso ***rerun*** ao job.

Sintaxe

{**rerun** | **rr**} *jobselect*
[**;from=**[*wkstat***#**]*job*
[**;at=***time*]
[**;pri=***pri*]]
[**;step=***step*]
[**;noask**]

Argumentos

jobselect
Especifica o nome de um ou mais jobs. Caracteres curinga são permitidos.

from=[*wkstat***#]***job*
Especifica o nome de uma tarefa definida no banco de dados cujo arquivo de tarefa ou comando será executado no lugar da tarefa especificada por *jobselect*.

*wkstat***#**
Especifica o nome da estação de trabalho na qual o job **from** é executado. O padrão é a estação de trabalho na qual o **conman** está sendo executado.

job Especifica o nome da definição de tarefa **from**. Os seguintes tipos de nomes de job não são permitidos:
- Os nomes de jobs submetidos com o uso dos comandos **submit file** e **submit docommand**.
- Os nomes alias de jobs submetidos com o uso do comando **submit job**.

Para utilizar o argumento **from**, você deve ter acesso ao banco de dados a partir do computador no qual o **conman** está sendo executado

at=*time*
Especifica o horário de início da tarefa rerun, expressado da seguinte forma:

*hhmm [**timezone** | **tz** tzname]* [**+***n* **days** | *date*]

em que:

hhmm A hora e minuto.

+*n* **days**
A próxima ocorrência de *hhmm* em *n* número de dias.

date A próxima ocorrência de *hhmm* em *date*, expressa como *mm/dd[/yy]*.

timezone | tz *tzname*
O nome do fuso horário do job. Consulte Capítulo 16, "Gerenciando fusos horários", na página 597 para obter nomes válidos.

pri=*pri*
Especifica a prioridade a ser atribuída ao job de reexecução. Se você não especificar uma prioridade, o job receberá a mesma prioridade do job original.

step=*step*
Especifica que o job é reexecutado utilizando esse nome no lugar do nome de job original. Consulte "Notas de Uso" para obter informações adicionais.

noask Especifica para não solicitar confirmação antes da tomada de ação em cada job qualificado.

Comentários

É possível reexecutar jobs que estão no estado SUCC, FAIL ou ABEND. Um job de reexecução é colocado no mesmo fluxo de job do job original e herda as dependências do job original. Se você reexecutar um job repetitivo (**every**), o job de reexecução será planejado para ser executado com a mesma taxa do job original.

Nota: É possível emitir o comando **rerun** para tarefas no fluxo de tarefas `EXTERNAL` que estejam no estado ERROR. As tarefas do fluxo de tarefas `EXTERNAL` representam tarefas e fluxos de tarefas que foram especificados como dependências da interligação de redes. O estado do job é definido inicialmente como **extrn** imediatamente após a execução de um comando **rerun** e o**conman** começa a verificar o estado.

Quando**;from** for usado, o nome da tarefa reexecutada dependerá do valor da Opção Global **enRetainNameOnRerunFrom**. Se a opção for definida como **Y**, os jobs de reexecução reterão os nomes de job originais. Se a opção for definida como **N**, os jobs de reexecução receberão os nomes de job **from**. Para obter informações adicionais, consulte o *Guia de Administração* do Tivoli Workload Scheduler.

Em exibições do **conman**, os jobs reexecutados são exibidos com a notação **>>rerun as**. Para referir-se a um job reexecutado em outro comando, tal como **altpri**, você deve utilizar o nome de job original.

Quando uma tarefa é reexecutada com a opção **;step**, a tarefa é executada com *step* no lugar de seu nome original. Em um script de job, é possível utilizar o comando **jobinfo** para retornar o nome do job e executar o script de forma diferente para cada iteração. Por exemplo, no script UNIX a seguir, o comando **jobinfo** é utilizado para definir uma variável chamada **STEP** com o nome utilizado para executar o job. A variável **STEP** é então utilizada para determinar como o script será executado.

```
...
MPATH=`maestro`
STEP=`$MPATH/bin/jobinfo job_name`
if [$STEP = JOB3]
  then
  ...
  STEP=JSTEP1
  fi
if [$STEP = JSTEP1]
  then
  ...
  STEP=JSTEP2
  fi
if [$STEP = JSTEP2]
  then
  ...
  fi
...
```

Em exibições do **conman**, os jobs reexecutados com a opção **;step** são exibidos com a notação **>>rerun step**.

Para obter informações sobre **jobinfo**, consulte "jobinfo" na página 511.

Exemplos

Para reexecutar o job `job4` no fluxo de job `sked1` na estação de trabalho`main`, execute o seguinte comando:

```
rr main#sked1.job4
```

Para executar novamente a tarefa `job5` no fluxo de tarefas `sked2` usando a definição de tarefa para a tarefa `jobx`, em que a hora **at** da tarefa está configurada para 18h30 e sua prioridade está configurada para **25**, execute o comando a seguir:

```
rr sked2.job5;from=jobx;at=1830;pri=25
```

Para reexecutar o job `job3` no fluxo de job `sked4` utilizando o nome de job `jstep2`, execute o seguinte comando:

```
rr sked4.job3;step=jstep2
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Monitorar Tarefas**.
2. Selecione **Todas as Tarefas no plano** ou outra tarefa para monitorar tarefas.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. Na tabela que contém a lista de tarefas, selecione uma tarefa e clique em **Mais Ações** > **Executar Novamente...**

resetFTA

Gera um arquivo `Sinfonia` atualizado e envia o mesmo a um agente tolerante a falhas no qual o arquivo `Symphony` foi corrompido.

Nota: A remoção completa e a substituição do arquivo Symphony causa alguma perda de dados como, por exemplo, os eventos sobre o status da tarefa ou o conteúdo das filas de mensagens Mailbox.msg e tomaster.msg. Se as informações de estado sobre uma tarefa estavam contidas nessas filas, essa tarefa será executada novamente. É recomendada a aplicação deste comando com cuidado.
No processo, os arquivos a seguir são movidos para o diretório *TWA_home*/TWS/tmp:

- Appserverbox.msg
- clbox.msg
- Courier.msg
- Intercom.msg
- Mailbox.msg
- Monbox.msg
- Moncmd.msg
- Symphony
- Sinfonia

Antes do comando ser executado, uma mensagem de informação é exibida para solicitar confirmação e assegurar que o comando não seja emitido por engano. Se um dos arquivos de destino não puder ser movido por estar sendo usado por outro processo (por exemplo, o processo mailman ainda está em execução), a operação não é executada e uma mensagem de erro é exibida.

Autorização

Você deve ter acesso **RESETFTA** aoagente tolerante a falhas que deseja reconfigurar.

Sintaxe

resetFTA cpu

Argumentos

cpu É oagente tolerante a falhas a ser reconfigurado.

Esse comando não está disponível no Dynamic Workload Console.

Exemplos

Para reconfigurar oagente tolerante a falhas denominado omaha, execute o comando a seguir:

```
resetFTA omaha
```

Consulte também

Para obter informações sobre o procedimento de recuperação do agente tolerante a falhas, consulte a seção sobre o procedimento de recuperação de um agente tolerante a falhas em *Tivoli Workload Scheduler: Guia de Resolução de Problemas*..

resource

Altera o número de unidades totais de um recurso.

Você deve ter acesso ***resource*** ao recurso.

Sintaxe

{**resource** |
reso} [*workstation***#**]
*resource***;***num*
[**;noask**]

Argumentos

estação de trabalho
Especifica o nome da estação de trabalho na qual o recurso foi definido. O padrão é a estação de trabalho na qual o **conman** está sendo executado.

resource
Especifica o nome do recurso.

num Especifica o número total de unidades de recurso. Os valores válidos são **0** a **1024**.

noask Especifica para não solicitar confirmação antes da tomada de ação em cada recurso qualificado.

Exemplos

Para alterar o número de unidades do recurso`otapes` para **5**, execute o seguinte comando:

```
resource tapes;5
```

Para alterar o número de unidades do recurso`jobslots` na estação de trabalho`site2` para **23**, execute o seguinte comando:

```
reso
site2#jobslots;23
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Monitorar Recursos**.
2. Selecione **Todos os Recursos no Plano** ou outra tarefa para monitorar recursos.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. Na tabela de resultados, selecione um recurso e clique em **Alterar Unidades...**

setsym

Seleciona um arquivo archive do plano de produção. Comandos de exibição subseqüentes mostram o conteúdo do plano de produção arquivado. Você não pode modificar as informações de um arquivo archive do plano de produção.

Sintaxe

{**setsym** | **set**} [**trial** | **forecast**] [*filenum*]

Argumentos

trial Lista planos trial.

previsão
Lista planos de previsão.

filenum
Especifica o número do arquivo archive do plano de produção. Se você não especificar um número de arquivo de log, o ponteiro retornará para zero, o plano de produção atual (Symphony). Utilize o comando **listsym** para listar números do arquivo archive.

Exemplos

Para selecionar o arquivo archive do plano de produção **5**, execute o seguinte comando:

```
setsym 5
```

Para selecionar o plano de produção atual (arquivo Symphony), execute o seguinte comando:

```
set
```

Consulte também

No Dynamic Workload Console:

1. Na barra de navegação, clique em **Planejando** > **Previsão de Carga de Trabalho** > **Gerenciar Planos Disponíveis**.
2. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
3. Clique em **Planos Arquivados** ou forneça um nome do arquivo de plano.
4. Clique em **Exibir Lista de Planos**.

showcpus

Exibe informações sobre estações de trabalho e links.

As informações exibidas são atualizadas apenas enquanto oTivoli Workload Scheduler (batchman) está em execução nas estações de trabalho. Se **batchman** está ativo ou inativo é confirmado na tela pela mensagem `Batchman LIVES` ou `Batchman down` quando você emite o comando `conman start`.

Você deverá ter acesso ***list*** para o objeto que está sendo mostrado se a opção ***enListSecChk*** tiver sido configurada como **yes** no gerenciador de domínio principal quando o plano de produção foi criado ou estendido.

Sintaxe

{**showcpus** | **sc**} [[*domain***!**]*workstation*]
[;info|;link]
[;offline]

{**showcpus** | **sc**} [[*domain***!**]*workstation*] **[;getmon]**

Argumentos

domínio
Especifica o nome de um domínio. O padrão é o domínio no qual o comando é executado.

estação de trabalho
Especifica o nome de uma estação de trabalho. O padrão é a estação de trabalho na qual o comando é executado. Quando não há especificação do domínio e da estação de trabalho, a saída pode ser a seguinte:

- O seguinte comando exibe todas as estações de trabalho que estão no domínio da estação de trabalho na qual o comando foi executado, além de todos os gerenciadores de domínio conectados se a estação de trabalho for um gerenciador de domínio.

  ```
  conman "sc"
  ```

- O seguinte comando exibe todas as estações de trabalho que estão no domínio da estação de trabalho na qual o comando foi executado, sem os gerenciadores de domínio conectados.

  ```
  conman "sc @"
  ```

info Exibe informações no formato **info**.

link Exibe informações no formato **link**.

off-line
Envia a saída do comando para o dispositivo de saída do **conman**. Para obter informações sobre esse dispositivo, consulte "Saída Off-line" na página 340.

getmon
Retorna a lista de regras de eventos definidas para o monitor em execução na estação de trabalho especificada no seguinte formato:

```
<rule_name>::<eventProvider>#<eventType>:<scope>
```

O escopo da regra são informações geradas automaticamente sobre atributos da regra, tais como as estações de trabalho em que ela é utilizada, um nome de tarefa ou arquivo, etc.

O cabeçalho da saída também contém o registro de data e hora de quando o pacote de configuração da regra foi gerado pela última vez.

Nota: Esta opção não é válida em estações de trabalhoTivoli dynamic workload broker (ou gerenciadores de domínio dinâmicos). Nesse caso, é possível recuperar as informações sobre as regras ativas definidas nessas estações de trabalho no arquivo *TWA_home*`\TWS\monconf\TWSObjectsMonitor.cfg` no gerenciador de domínio principal.

Resultados

Quando o parâmetro`getmon` não é utilizado, a saída do comando é produzida nos três formatos, **standard**, **info** e **link**. O valor padrão é **standard**. O significado dos caracteres exibidos depende do tipo de formato selecionado.

Ao executar em uma estação de trabalho com uma versão do Tivoli Workload Scheduler anterior à versão 8.6, o comando **sc** mostra como FTA os tipos de estações de trabalho introduzidos com o Tivoli Workload Scheduler versão 8.6, conjunto, conjunto dinâmico, agente e mecanismo remoto.

Quando o parâmetro`getmon` é utilizado, a lista de regras é fornecida como uma saída separada.

Exemplos

1. Para exibir informações sobre a estação de trabalho na qual o **conman** está sendo executado no formato **info**, execute o seguinte comando:

```
showcpus ;info
```

Uma saída de amostra para este comando é:

```
CPUID   VERSION TIME ZONE      INFO
MASTER   8.6.0.0 US/Pacific    Linux 2.6.5-7.191-s390 #1 SM
FTA1   8.6.0.0       Linux 2.4.9-e.24 #1 Tue May
FTA2   8.6.0.0       HP-UX B.11.11 U 9000/785
```

2. Para exibir as informações de link para todas as estações de trabalho, execute o seguinte comando:

```
sc @!@;link
```

Uma saída de amostra é a seguinte:

```
CPUID  HOST  FLAGS ADDR NODE
MASTER  MASTER  AF T 51099 9.132.239.65
FTA1  FTA1  AF T 51000 CPU235019
FTA2  FTA2  AF T 51000 9.132.235.42
BROKER1 MASTER  A T 51111 9.132.237.17
```

3. Para exibir informações sobre a estação de trabalho, execute o seguinte comando:

```
showcpus
```

Se você executar este comando em um ambiente quando a conexão primária da estação de trabalho com seu gerenciador de domínio ou superior não estiver ativa, receberá a seguinte saída:

```
CPUID  RUN  NODE  LIMIT FENCE DATE TIME STATE METHOD DOMAIN
MASTER 360 *WNT MASTER 10 0 03/05/2010 1348 I J E   MASTERDM
FTA1 360 WNT FTA  10 0 03/05/2010 1348 FTI JW M   MASTERDM
FTA2 360 WNT FTA  10 0 03/05/2010 1348 FTI JW M   MASTERDM
FTA3 360 WNT MANAGER 10 0 03/05/2010 1348 LTI JW M   DOMAIN1
FTA4 360 WNT FTA  10 0 03/05/2010 1348 F I J M   DOMAIN1
FTA5 360 WNT FTA  10 0 03/05/2010 1348 I J M   DOMAIN1
SA1  360 WNT S-AGENT 10 0 03/05/2010 1348 F I J M   DOMAIN1
XA_FTA4 360 OTHR X-AGENT 10 0 03/05/2010 1348 L I J M   DOMAIN1
FTA6 360 WNT MANAGER 10 0 03/05/2010 1348 F I J M   DOMAIN2
FTA7 360 WNT FTA  10 0 03/05/2010 1349 F I J M   DOMAIN2
FTA7 360 WNT FTA  10 0 03/05/2010 1349 F I J M   DOMAIN2
BROKER 360 OTHR BROKER 10 0 03/05/2010 1349 LTI JW    MASTERDM
```

Se você executar este comando em um ambiente quando a conexão primária da estação de trabalho com seu gerenciador de domínio ou superior estiver ativo e pelo menos uma conexão secundária não estiver ativa, receberá a seguinte saída:

```
CPUID  RUN  NODE  LIMIT FENCE DATE TIME STATE METHOD DOMAIN
MASTER 360 *WNT MASTER 10 0 03/05/2010 1348 I J E   MASTERDM
FTA1 360 WNT FTA  10 0 03/05/2010 1348 FTI JW M   MASTERDM
FTA2 360 WNT FTA  10 0 03/05/2010 1348 FTI JW M   MASTERDM
FTA3 360 WNT MANAGER 10 0 03/05/2010 1348 FTI JW M   DOMAIN1
FTA4 360 WNT FTA  10 0 03/05/2010 1348 F I J M   DOMAIN1
FTA5 360 WNT FTA  10 0 03/05/2010 1348 L I M   DOMAIN1
SA1  360 WNT S-AGENT 10 0 03/05/2010 1348 F I J M   DOMAIN1
XA_FTA4 360 OTHR X-AGENT 10 0 03/05/2010 1348 L I J M   DOMAIN1
FTA6 360 WNT MANAGER 10 0 03/05/2010 1348 F I J M   DOMAIN2
FTA7 360 WNT FTA  10 0 03/05/2010 1349 F I J M   DOMAIN2
```

Se você executar este comando em um ambiente quando a conexão primária da estação de trabalho com seu gerenciador de domínio ou superior e todas as conexões secundárias estiverem ativas, receberá a seguinte saída:

```
CPUID  RUN  NODE  LIMIT FENCE DATE TIME STATE METHOD DOMAIN
MASTER 360 *WNT MASTER 10 0 03/05/2010 1348 I J E   MASTERDM
FTA1 360 WNT FTA  10 0 03/05/2010 1348 FTI JW M   MASTERDM
FTA2 360 WNT FTA  10 0 03/05/2010 1348 FTI JW M   MASTERDM
FTA3 360 WNT MANAGER 10 0 03/05/2010 1348 FTI JW M   DOMAIN1
FTA4 360 WNT FTA  10 0 03/05/2010 1348 F I J M   DOMAIN1
```

```
FTA5 360 WNT FTA  10 0 03/05/2010 1348 F I M   DOMAIN1
SA1  360 WNT S-AGENT 10 0 03/05/2010 1348 F I J M   DOMAIN1
XA_FTA4 360 OTHR X-AGENT 10 0 03/05/2010 1348 L I J M   DOMAIN1
FTA6 360 WNT MANAGER 10 0 03/05/2010 1348 F I J M   DOMAIN2
FTA7 360 WNT FTA  10 0 03/05/2010 1349 F I J M   DOMAIN2
```

4. Obtenha uma lista de monitores de regras ativas na estação de trabalho denominada CPU1, execute este comando:

```
sc CPU1 getmon
```

   Você obterá a seguinte saída:

```
Monitoring configuration for CPU1:
*****************************************
*** Package Date : 04/22/2009 12:00 GMT ***
*****************************************
Rule1::FileMonitor#FileCreated:Workstation=CPU1,CPU2;File="\tmp\filename"
Rule2::FileMonitor#ModificationCompleted:Workstation=CPU1,CPU3;File="\staging\orders"
Rule3::TWSObjectsMonitor#JobSubmit:JobKey=CPU1#JS1.Job1
Rule5::TWSObjectsMonitor#JobLate:JobKey=CPU1#JS1.Job1
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Ambiente** > **Monitorar Estações de Trabalho**.
2. Selecione uma tarefa para monitorar estações de trabalho.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.

Formato Padrão

CPUID
O nome da estação de trabalho na qual essa informação se aplica.

RUN O número de execuções do arquivo Symphony.

NODE
O tipo de nó e o tipo de estação de trabalho. Os tipos de nós são os seguintes:
- UNIX
- WNT
- OUTRO
- ZOS
- IBM i

Os tipos de estação de trabalho são os seguintes:
- MASTER
- MANAGER
- FTA
- S-AGENT
- X-AGENT
- AGENT
- POOL
- D-POOL
- REM-ENG

LIMIT
O limite de tarefas do Tivoli Workload Scheduler.

FENCE
O limitador de tarefas do Tivoli Workload Scheduler.

DATE TIME

A data e hora em que o Tivoli Workload Scheduler começou a executar o plano de produção atual (arquivo `Symphony`).

STATE

Exibe as seguintes informações:

- O estado dos links e processos da estação de trabalho. São exibidos até cinco caracteres da seguinte forma: A explicação dos caracteres é dividida com base no escopo do caractere:

  ```
  [L|F] [T|H|X|B] [I] [J] [W|H|X] [M] [E|e] [D] [A|R]
  ```

  em que:

  L O link principal é aberto (vinculado) a seu domínio ou gerenciador superior.

  Se a estação de trabalho for do tipoagent ou mecanismo remoto, este sinalizador indica que a estação de trabalho está conectada ao servidor broker de carga de trabalho.

  Se a estação de trabalho for do tipo conjunto ou conjunto dinâmico, esse sinalizador indica que a estação de trabalho do Workload Broker na qual o conjunto ou conjunto dinâmico está registrado está vinculada a seu domínio ou gerenciador superior.

  D A estação de trabalho está totalmente vinculada por meio de conexões principais e de todas as secundárias. Esse sinalizador aparecerá somente se a opção global ***enSwfaultTol*** estiver configurada para *YES* utilizando a linha de comandos **optman** no gerenciador de domínio principal e indica que a estação de trabalho está diretamente vinculada a seu gerenciador de domínio e a todos os seus gerenciadores de domínio de backup completo. Para obter informações sobre como utilizar a linha de comandos **optman**, consulte o *Guia de Administração do IBM Tivoli Workload Scheduler*.

  P Esse sinalizador será exibido se oagente tolerante a falhas estiver vinculado diretamente ao gerenciador de domínio a partir do qual o comando foi executado.

  U A estação de trabalho está ligada através de seu host.

  X A estação de trabalho está ligada como um agente estendido (x-agent).

  B A estação de trabalho se comunica por meio do servidor do Workload Broker.

  I Se a estação de trabalho for do tipo agente MASTER, MANAGER, FTA, S-AGENT, X-AGENT, esse sinalizador indica que o programa **jobman** concluiu a inicialização.

  Se a estação de trabalho for do tipoagent, conjunto ou conjunto dinâmico, este sinalizador indicará que o agente foi inicializado corretamente.

  Se a estação de trabalho for do tipo mecanismo remoto, esse sinalizador indica que a comunicação entre a estação de trabalho do mecanismo remoto e a mecanismo remoto está inicializada corretamente.

  J Se a estação de trabalho for do tipo agente MASTER, MANAGER, FTA, S-AGENT, X-AGENT, esse sinalizador indica que o programa **jobman** está em execução.

  Se a estação de trabalho for do tipoagent, este sinalizador indicará que o JobManager está em execução. Como nenhum

monitoramento é executado nas estações de trabalho de conjunto dinâmico, para esse tipo de estação de trabalho, o caractere J sempre é mostrado.

Se a estação de trabalho for do tipo conjunto, esse sinalizador indica que o processo JobManager está em execução em pelo menos um agente registrado no conjunto.

Se a estação de trabalho for do tipo mecanismo remoto, esse sinalizador indica que o comando ping para o mecanismo remoto foi bem-sucedido.

W A estação de trabalho é vinculada via TCP/IP usando o processo de **gravação**.

Se a estação de trabalho executando**conman** estiver diretamente vinculada à estação de trabalho remota, você verá o sinalizador W por causa do mailman local estar vinculado ao processo de gravação remoto.

```
LTI JW
```

Se a estação de trabalho executando**conman** não estiver diretamente vinculada à estação de trabalho remota, você não verá o sinalizador W por causa do mailman local não estar diretamente vinculado ao processo de gravação remoto.

```
L I J
```

Para obter detalhes adicionais sobre o processo **writer**, o tópico sobre processos de rede no *Tivoli Workload Scheduler: Administration Guide*.

Nota: Se a estação de trabalho que está executando**conman** for o host do agente estendido, o estado do agente estendido é

```
LXI JX
```

Se a estação de trabalho que está executando**conman** não for o host do agente estendido, o estado do agente estendido é

```
LHI JH
```

- O estado do agente de monitoramento. Até três caracteres são exibidos, conforme a seguir:

```
[M] [E|e] [D]
```

em que:

S O processo `monman` está em execução. Esse sinalizador é exibido para todas as estações de trabalho da rede quando o recurso de automação de carga de trabalho orientado por evento está ativado (a opção global `enEventDrivenWorkloadAutomation` está configurada para `yes`), com a exceção das estações de trabalho em que `monman` foi parado manualmente (usando**conman** ou Dynamic Workload Console).

E O servidor de processamento de eventos está instalado e em execução na estação de trabalho.

J O servidor de processamento de eventos está instalado na estação de trabalho mas não está em execução.

D A estação de trabalho está utilizando uma configuração de monitoramento de pacotes atualizada. Esse sinalizador é exibido para as estações de trabalho nas quais o pacote mais recente de regras de eventos foi implementado (manualmente com o comando `planman deploy` ou automaticamente com a frequência especificada pela opção global `deploymentFrequency`).

- O estado do WebSphere Application Server. Um sinalizador de um caractere é exibido, se o servidor de aplicativos estiver instalado:

  ```
  [A|R]
  ```

  em que:

  A O WebSphere Application Server foi iniciado.

  R O WebSphere Application Server está reiniciando.

  O sinalizador estará em branco se o servidor de aplicativos estiver inativo ou se ele não foi instalado.

METHOD
O nome do método de acesso especificado na definição de estação de trabalho. Apenas para agentes estendidos.

DOMAIN
O nome do domínio do qual a estação de trabalho é um membro.

Formato Info

CPUID
O nome da estação de trabalho na qual essa informação se aplica.

VERSION
A versão do agente Tivoli Workload Scheduler instalada na estação de trabalho.

TIMEZONE
O fuso horário da estação de trabalho. É o mesmo valor da variável de ambiente TZ. Para um agente estendido, esse é o fuso horário de seu host. Para uma estação de trabalho de mecanismo remoto, esse é o fuso horário do mecanimso remoto.

INFO Um campo informativo. Para todos os tipos de estação de trabalho, exceto o agente estendido e as estações de trabalho do intermediário, ele contém a versão do sistema operacional e o modelo de hardware. Para agentes estendidos e estações de trabalho de mecanismo remoto, nenhuma informação está listada. Para a estação de trabalho de mecanismo remoto, mostra Mecanismo Remoto.

Formato Link

CPUID
O nome da estação de trabalho na qual essa informação se aplica.

HOST O nome da estação de trabalho atuando como o host para um agente padrão ou agente estendido. Para gerenciadores de domínio e agente tolerante a falhas, é o mesmo de CPUID. Para as estações de trabalho do agente padrão e do intermediário, é o nome do gerenciador de domínio. Para agentes estendidos, é o nome do estação de trabalho host.

FLAGS
O estado das propriedades da estação de trabalho. São exibidos até cinco caracteres, como a seguir:

```
[A] [B] [F] [s] [T]
```

A O link automático está ativado na definição de estação de trabalho.

B Esse sinalizador será utilizado somente em ambientes de ponta a ponta e isso indica se o sinalizador **desativar ativação de job** está desativado.

D O modo Status Completo está ativado na definição de estação de trabalho.

s O ID do servidor **mailman** da estação de trabalho.
P O link está definido como TCP/IP.

ADDR
O número da porta TCP/IP da estação de trabalho.

NODE
O nome do nó da estação de trabalho.

showdomain

Exibe informações sobre o domínio.

As informações exibidas são atualizadas somente se oTivoli Workload Scheduler (batchman) estiver em execução. A confirmação da atividade ou inatividade do batchman aparece na tela através da mensagem `Batchman LIVES` ou `Batchman down` quando você emite o comando `conman start`.

Você deverá ter acesso ***list*** para o objeto que está sendo mostrado se a opção ***enListSecChk*** tiver sido configurada como **yes** no gerenciador de domínio principal quando o plano de produção foi criado ou estendido.

Sintaxe

{**showdomain** | **showd**} [*domain*]
[**;info**]
[**;offline**]

Argumentos

domínio
Especifica o nome do domínio. O padrão é o domínio no qual o **conman** está sendo executado. Caracteres curinga são permitidos.

info Exibe informações no formato **info**.

off-line
Envia a saída do comando para o dispositivo de saída do **conman**. Para obter informações sobre esse dispositivo, consulte "Saída Off-line" na página 340.

Resultados

A saída do comando é produzida em dois formatos: **padrão** e **info**.

Exemplos

Para exibir informações sobre o domínio`masterdm`, execute o seguinte comando:

```
showdomain masterdm
```

Uma saída de amostra é a seguinte:

```
DOMAIN   MANAGER   PARENT
*MASTERDM  *MASTER
```

Para exibir as estações de trabalho membros em todos os domínios no formato **info**, execute o seguinte comando:

```
showdomain @;info
```

uma saída de amostra é a seguinte:

```
DOMAIN   MEMBER-CPUs  CPU-Type
 MASTERDM  *MASTER   MASTER
 DOM1    FTA1    MANAGER
 DOM2    FTA2    MANAGER
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Ambiente** > **Monitorar Domínios**.
2. Selecione uma tarefa para monitorar domínios.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.

Formato Padrão

DOMAIN
O nome do domínio no qual essa informação se aplica.

MANAGER
O nome do gerenciador de domínio.

PARENT
O nome do domínio pai.

Formato Info

DOMAIN
O nome do domínio no qual essa informação se aplica.

MEMBER-CPUS
Os nomes das estações de trabalho do domínio.

CPU-TYPE
O tipo de cada estação de trabalho: MASTER, MANAGER, FTA, S-AGENT, X-AGENT ou BROKER.

showfiles

Exibe informações sobre dependências de arquivos. Uma dependência file ocorre quando um job ou fluxo de job depende da existência de um ou mais arquivos para poder começar a execução.

As informações exibidas são atualizadas somente se oTivoli Workload Scheduler (batchman) estiver em execução. A confirmação da atividade ou inatividade do batchman aparece na tela através da mensagem `Batchman LIVES` ou `Batchman down` quando você emite o comando `conman start`.

Sintaxe

{**showfiles** | **sf**} [[*workstation*#]*file*]
[**;***state*[**;**...]]
[**;keys**]
[**;offline**]

{**showfiles** | **sf**} [[*workstation*#]*file*]
[**;***state*[**;**...]]
[**;deps**[**;keys** | **info** | **logon**]]
[**;offline**]

Argumentos

estação de trabalho
Especifica o nome da estação de trabalho na qual o arquivo existe. O padrão é a estação de trabalho na qual o **conman** está sendo executado. Caracteres curinga são permitidos.

file Especifica o nome do arquivo. O nome deverá ser colocado entre aspas (") se contiver caracteres diferentes do seguinte: alfanuméricos, traços (-), barras (/), barras invertidas (\) e sublinhados (_). O padrão é exibir todas as dependências de arquivos. Caracteres curinga são permitidos.

state Especifica o estado das dependências de arquivos a serem exibidas. O padrão é exibir dependências de arquivos em todos os estados. Os estados são os seguintes:

sim O arquivo existe e está disponível.

não O arquivo não está disponível ou não existe.

? A disponibilidade está sendo verificada.

<branco>
O arquivo ainda não foi verificado ou estava disponível e sendo utilizado para satisfazer uma dependência de job ou de fluxo de job.

keys Exibe uma única coluna de lista dos objetos selecionados pelo comando.

deps Exibe informações no formato **deps**. Utilize **keys**, **info** ou **logon** para modificar a exibição.

off-line
Envia a saída do comando para o dispositivo de saída do **conman**. Para obter informações sobre esse dispositivo, consulte "Saída Off-line" na página 340.

Resultados

A saída do comando é produzida em três formatos: **padrão**, **keys** e **deps**. Os argumentos **keys**, **info** e **logon** modificam a exibição de **deps**.

Exemplos

Para exibir o status de uma dependência de arquivo para `d:\apps\mis\lib\data4`, execute o seguinte comando:

```
showfiles d:\apps\mis\lib\data4
```

Para exibir o status **off-line** de todas as dependências de arquivo em todas as estações de trabalho no formato `deps`, execute o seguinte comando:

```
sf @#@;deps;offline
```

Para exibir o status de todas as dependências de arquivos em todas as estações de trabalho no formato `deps`, execute o seguinte comando:

```
sf @#@;deps
```

Uma saída de amostra é a seguinte:

```
                                          (Est) (Est)
Workstation Job Stream SchedTime Job   State Pr Start Elapse ReturnCode Dependencies

MASTER#/test/^LFILEJOB^ Dependencies are:
 MASTER  #LFILEJOB 0600 11/26 ******** READY 10
```

```
                                LFILEJOB  HOLD 10 (11/26)     ^LFILEJOB^

MASTER#/usr/home/me10_99/`/usr/home/me10_99/bin/parms FILE_JS1` Dependencies are:
 MASTER  #FILE_JS1 0600 11/26 ******** HOLD 10 (11/26)     parms FILE_JS1`
                                FILE_JS1 HOLD 10 (11/26)

MASTER#/usr/home/me10_99/`/usr/home/me10_99/bin/parms FILE_JOB1` Dependencies are:
 MASTER  #FILE_JOB1 0600 11/26 ******** READY 10
                                FILE_JB1 HOLD 10 (11/26)     parms FILE_JB1`
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Monitorar Arquivos**.
2. Selecione uma tarefa para monitorar arquivos.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.

Formato Padrão

Existe O estado da dependência de arquivo.

Nome do Arquivo
O nome do arquivo.

Formato Keys

Os arquivos são listados um em cada linha. Os nomes dos diretórios não são incluídos. Cada arquivo é listado no seguinte formato:

```
workstation#file
```

Formato Deps

Os arquivos são listados seguidos dos jobs e fluxos de job dependentes. Os jobs são listados no formato **showjobs.** Os fluxos de job são listados no formato **showschedules** padrão.

Formato Deps;keys

Os jobs e os fluxos de job que possuem dependências `file` são listados um em cada linha, no seguinte formato:

```
workstation#jstream[.job]
```

Formato Deps;info

Os arquivos são listados seguidos dos jobs e fluxos de job dependentes. Os jobs são listados no formato **showjobs;info.** Os fluxos de job são listados no formato **showschedules** padrão.

Formato Deps;logon

Os arquivos são listados seguidos dos jobs e fluxos de job dependentes. Os jobs são listados no formato **showjobs;logon**. Os fluxos de job são listados no formato **showschedules** padrão.

showjobs

Exibe informações sobre jobs.

As informações exibidas são atualizadas somente se oTivoli Workload Scheduler (batchman) estiver em execução. A confirmação da atividade ou inatividade do

batchman aparece na tela através da mensagem `Batchman LIVES` ou `Batchman down` quando você emite o comando `conman start`.

Você deverá ter acesso ***list*** para o objeto que está sendo mostrado se a opção ***enListSecChk*** tiver sido configurada como **yes** no gerenciador de domínio principal quando o plano de produção foi criado ou estendido.

Sintaxe

{**showjobs** | **sj**} [*jobselect*]
[**;keys** | **info** | **step** | **logon** | **crit** | **keys retcod**]
[**;short** | **single**]
[**;offline**]
[**;showid**]

{**showjobs** | **sj**} [*jobselect*]
[**;deps**[**;keys** | **info** | **logon**]]
[**;short** | **single**]
[**;offline**]
[**;showid**]
[**;props**]

{**showjobs** | **sj**} [*jobselect* |
[**workstation#**]*jobnumber.hhmm*]
[**;stdlist**[**;keys**]]
[**;short** | **single**]
[**;offline**]
[**;showid**]
[**;props**]

Argumentos

crit Exibe informações no formato **crit**.

deps Exibe informações no formato deps; ou seja, as tarefas usadoa em dependências de *sequências* são listadas seguidas pelas tarefas e fluxos de tarefas dependentes. As tarefas são listadas no formato `showjobs` básico. Os fluxos de tarefas são listados no formato `showschedules` básico. Use "keys", "info" ou "logon" para modificar a exibição de "deps".

hhmm A hora em que o job foi iniciado. Utilize essa opção junto com os argumentos **stdlist** e **single**, para exibir uma determinada ocorrência do job.

info Exibe informações no formato **info**.

jobnumber
O número do job.

jobselect
Consulte “Selecionando Tarefas em Comandos” na página 345.

keys Exibe uma única coluna de lista dos objetos selecionados pelo comando.

logon Exibe informações no formato **logon**.

off-line
Envia a saída do comando para o dispositivo de saída do **conman**. Para obter informações sobre esse dispositivo, consulte “Saída Off-line” na página 340.

props Exibe as informações a seguir sobre a instância da tarefa especificada. Você deve ter acesso de exibição às propriedades da instância da tarefa especificada que estiver sendo mostrada:

Informações Gerais

- Tarefa
- Estação de Trabalho
- Tarefa
- Tipo de Tarefa
- Fluxo de Tarefas
- Estação de Trabalho do Fluxo de Tarefas
- Tempo Planejado
- Prioridade
- Efetuar Login
- Monitorado
- Requer Confirmação
- Interactive (Interativo)
- Crítico

Informações sobre o Tempo de Execução

- Status
- Status Interno
- Dependências Não Satisfeitas
- Número da Tarefa
- Opções para Executar Novamente
- Informações
- Promovido
- Código de Retorno
- Expressão de Mapeamento de Código de Retorno

Informações de Tempo

- Início Real
- Início Mais Antecipado
- Último Início
- Ação de Início mais Recente
- Duração Máxima
- Ação de Duração Máxima
- Duração Mínima
- Ação de Duração Mínima
- Último Início Crítico
- Prazo final
- Repetir Intervalo
- Duração Real
- Duração Estimada

Informações de Recuperação

- Ação
- Mensagem

- Definição de Tarefa
- Estação de Trabalho

Extra Information

Esta seção mostra propriedades adicionais específicas para tarefas sombra e tarefas definidas por JSDL. Para tarefas sombra, contém as seguintes informações:

Para tarefas sombra distribuídas:

- Horário Plaejado da Tarefa Remota
- Tarefa Remota
- Fluxo de Tarefas Remoto
- Estação de Trabalho do Fluxo de Tarefas Remoto

Para tarefas sombra do z/OS:

- Horário Plaejado da Tarefa Remota
- Tarefa Remota
- Estação de Trabalho de Tarefa Remota
- Código de Erro da Tarefa Remota

Para obter informações adicionais, consulte a "Como o Status da Tarefa Sombra Muda após a Conexão Ser Estabelecida" na página 637.

Nota: Informações sobre tarefas arquivadas não podem ser recuperadas usando a opção **props**.

retcod Exibe o código de retorno do job. Esse argumento deve ser utilizado em conjunto com o argumento **keys**, por exemplo:

```
%sj @; keys retcod
```

short Reduz a exibição dos jobs **every** e de **rerun** para incluir apenas o seguinte:

- A primeira iteração
- Jobs em estados diferentes
- Jobs exatamente correspondidos

Nota: Esse campo mostra as propriedades específicas se a tarefa for uma tarefa sombra ou uma tarefa definida por JSDL.

showid

Exibe para cada fluxo de tarefa o identificador do fluxo de tarefa.

único Seleciona apenas o job pai em uma cadeia que pode incluir jobs de reexecução, de repetição e de recuperação. O job deve ser identificado pelo número de job em *jobselect*. Isso é útil com a opção **stdlist**.

stdlist Exibe informações no formato **stdlist**. Utilize o argumento **keys** para modificar a exibição.

Nota: Informações sobre tarefas arquivadas não podem ser recuperadas usando a opção **stdlist**.

step Exibe informações no formato **step**.

estação de trabalho

O nome da estação de trabalho em que a tarefa é executada. Caracteres curinga são permitidos.

Resultados

A saída do comando **showjobs** é produzida em oito formatos: **padrão**, **keys**, **info**, **step**, **logon**, **deps**, **crit** e **stdlist**. Os argumentos **keys**, **info**, **crit** e **logon** modificam as exibições.

Exemplos

Para exibir o status de todas as tarefas no fluxo de tarefas `acctg` na estação de trabalho`site3`, será possível executar o comando `showjobs` em um destes dois formatos:

```
showjobs site3#acctg.@
```

ou:

```
showjobs site3#acctg
```

Para exibir o status da tarefa `JBA` que pertence ao fluxo de tarefas `TEST1_DOCOM` na estação de trabalho`CPUA`, na qual está executando**conman**, e solicitar a exibição do identificador do fluxo de tarefas para o fluxo de tarefas, execute o seguinte comando:

```
sj CPUA#TEST1(0900 02/19/06).JBA
```

Uma saída de amostra para este comando é a seguinte:

```
Workstation Job Stream SchedTime Job   State Pr Start Elapse ReturnCode Dependencies

CPUA  #TEST1 0900 02/19 *** HOLD 0(02/19)     {02/20/06}; -TEST-
                                JBA HOLD 66(14:30)     J2(0600 02/24/06).JB1
```

A dependência **at** é mostrada como `(14:30)` na coluna Iniciar e a dependência follows a partir da tarefa `J2(0600 02/24/06).JB1` para a tarefa `JOBA` é mostrada na coluna Dependências.

Na coluna Dependências, a data colocada entre colchetes, `{02/20/06}`, indica que a instância do fluxo de tarefas foi transportado adiante e a data indica o dia quando a instância do fluxo de tarefas foi incluído no plano de produção pela primeira vez.

Para exibir o status das tarefas que pertencem ao fluxo de tarefas `JSDOC` na estação de trabalho`site3`, na qual está executando**conman**, e solicitar a exibição do identificador do fluxo de tarefas para o fluxo de tarefas, execute o seguinte comando:

```
%sj JSDOC.@;showid
```

Uma saída de amostra para este comando é a seguinte:

```
Workstation Job Stream SchedTime Job   State Pr Start Elapse ReturnCode Dependencies

site3  #JSDOCOM 0600 11/26 *** SUCC 10 11/26 00:01   {0AAAAAAAAAAAACRZ}
                               JDOC SUCC 10 11/26 00:01  0 #J25565
```

O identificador do fluxo de tarefas `0AAAAAAAAAAAACRZ` para o fluxo de tarefas `JDOCOM` é mostrado na coluna Dependências.

Nota: A hora ou data exibida na coluna **Início** é convertida para o fuso horário configurado na estação de trabalho onde o fluxo de tarefa deve ser executado.

Para exibir o status das tarefas pertencentes ao fluxo de tarefas JSDOCOM na estação de trabalhosite3 e pedir para mostrar as informações sobre o ID do usuário sob o qual a tarefa é executada, execute o seguinte comando:

```
sj site3#JSDOCOM.@;logon
```

Uma saída de amostra para este comando é a seguinte:

```
Workstation Job Stream SchedTime  Job   State Job#  Logon ReturnCode
site3  #JSDOCOM 0600 11/26
                                  JDOCOM SUCC #J25565 me10_99  0
```

Para exibir o status de todos os jobs no estado HOLD em todas as estações de trabalho no formato **deps**, execute o seguinte comando:

```
sj @#@.@+state=hold;deps
```

uma saída de amostra é a seguinte:

```
Workstation Job Stream SchedTime Job State Pr Start Elapse RetCode Dependencies

CPUA#JS2.JOBB Dependencies are:


CPUA  #JS21  0900 02/19 ***** HOLD 0(02/19)     {02/20/06}; -TEST-                JOBA HOLD 66(14:30)
                                                                                 JS22(0600 02/24/06).JOBB

CPUA#JS25.JOBC Dependencies are:


CPUA  #JS25  0600 02/24 ***** HOLD 10(02/24)     {02/20/06}
                              jobaa HOLD 10(02/24)(00:01)   TEST1; JOBC                              TEST2; JOBB
                                                                                 JS18(0600 02/24/06).@

CPUA#JS25.JOB1 Dependencies are:


CPUA  #JS25  0600 02/24 ***** HOLD 10(02/24)     {02/20/06}
                              JOBC HOLD 10(02/24)(00:01)   JOB1
                              jobaa HOLD 10(02/24)(00:01)   TEST1; JOBC                              TEST2; JOBB
JS18(0600 02/24/06).@
```

Para exibir o registro dos arquivos de lista padrão para a tarefa J25 no fluxo de tarefas JS25(0600 02/19/06) na estação de trabalhoCPUA, em execução em um ambiente UNIX, execute o seguinte comando:

```
sj CPUA#JS25(0600 02/19/06).J25;stdlist
```

A saída é a seguinte:

```
================================================================
= JOB  : CPUA#JS25[(0600 02/19/06),(0AAAAAAAAAAABQM)].J25
= USER  : tme10_99
= JCLFILE : ls
= Job Number: 28630
= Mon 02/20/06 07:57:37 PST
================================================================
Tivoli Workload Scheduler (UNIX)/JOBMANRC
AWSBIS307I Starting /usr/home/tme10_99/jobmanrc ls

Tivoli Workload Scheduler (UNIX)/JOBINFO 8.4 (9.5)
Installed for user "tme10_99".
Locale LANG set to the following: "en"
...
... <stdout, stderr, and echoed commands>
...
AWSBIS308I End of job
================================================================
= Exit Status : 0
= System Time (Seconds) : 0 Elapsed Time (Minutes) : 0
= User Time (Seconds) : 0
= Mon 02/20/06 07:57:38 PST
================================================================
```

em que:

Exit Status
É o status do job ao ser concluído.

Tempo Decorrido
É o tempo decorrido para o job.

System Time
É o tempo que o sistema kernel gastou para o job.

User Time
É o tempo que o usuário do sistema gastou para o job.

Nota: Os campos **System Time** e **User Time** serão utilizados somente no UNIX. Seus valores no Windows são sempre definidos como **0**. Isso é porque, no Windows, o processo **joblnch.exe** é executado em um tempo muito curto, que pode ser considerado nulo.

Para exibir as propriedades da tarefa sombra com o número de tarefa `546863237`, execute o comando a seguir:

```
sj 546863237;props
```

Uma saída de amostra para este comando é a seguinte:

```
Informações Gerais
  Job = D_SHADOW_JOB
  Workstation = REMENG1
  Task =
    <jsdl:jobDefinition
          xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
          xmlns:dshadow="http://www.ibm.com/xmlns/prod/scheduling/1.0/dshadow"
          xsi:schemaLocation="http://www.ibm.com/xmlns/prod/scheduling/1.0/dshadow">
       <jsdl:application name="distributedShadowJob">
          <dshadow:DistributedShadowJob>
             <dshadow:JobStream>JS1</dshadow:JobStream>
             <dshadow:Workstation>MYWKST</dshadow:Workstation>
             <dshadow:Job>JOBDEF1</dshadow:Job>
             <dshadow:matching>
                <dshadow:previous/>
             </dshadow:matching>
          </dshadow:DistributedShadowJob>
       </jsdl:application>
    </jsdl:jobDefinition>
  Task Type = distributedShadowJob
  Job Stream = JSDIST
  Job Stream Workstation = MYWKST
  Scheduled Time = 2010/08/11 06:00 TZ CEST
  Priority = 10
  Login =
  Monitored = No
  Requires Confirmation = No
  Interactive = No
  Critical = No
Informações sobre o Tempo de Execução
  Status = Undecided
  Internal Status = DONE
  Not Satisfied Dependencies = 0
  Job Number = 546863237
  Rerun Options =
  Information =
  Promoted = No
  Return Code =
  Return Code Mapping Expression =
Informações de Tempo
  Actual Start = 2010/08/11 12:00 TZ CEST
  Earliest Start =
  Latest Start =
  Latest Start Action =
  Maximum Duration =
  Maximum Duration Action =
  Minimum Duration = 000:01 (hhh:mm)
  Minimum Duration Action = Continue
  Critical Latest Start =
  Deadline =
  Repeat Range =
  Actual Duration =
  Estimated Duration = 00:02 (hh:mm)

Informações de Recuperação
  Action =
  Mensagem =
  Job Definition =
  Workstation =
Extra Information
  Remote Job Scheduled Time = 08/11/2010 06:00 TZ CEST
  Remote Job = JOBDEF1
  Remote Job Stream = JS1
  Remote Job Stream Workstation = MYWKST
```

O exemplo a seguir exibe o status do job `dbseload` com um código de retorno de **7** e um estado de SUCCESSFUL:

```
$ conman sj workstation#DAILY_DB_LOAD
Tivoli Workload Scheduler (UNIX)/CONMAN 8.4 (1.36.2.22) Materiais Licenciados -
Propriedade da IBM(R)
5698-WSH
(C) Copyright IBM Corp 1998, 2007 Todos os Direitos Reservados.
Direitos Restritos para Usuários do Governo dos Estados Unidos
Uso, duplicação ou divulgação restritos pelo documento
GSA ADP Schedule Contract com a IBM Corp.
IBM é uma marca registrada da International Business Machines
Corporation nos Estados Unidos e/ou em outros países.
Installed for user "tme10_99".
Locale LANG set to the following: "en"
Planejado para (Exp) 20/02/06 (#35) em CPUA. Batchman LIVES. Limit:50,Fence:0,Audit Level:0
Batchman ATIVO. Limit:50,Fence:0,Audit Level:0
sj workstation#DAILY_DB_LOAD
(Est) (Est)
CPU Schedule Job State Pr Start
Código de Retorno de Dependências Decorridas
WORKSTATION #DAILY_DB_LOAD ****************************** SUCC 10 22:11
00:04
DATASPLT SUCC 10 22:11
00:01 #J17922 0
DATAMRGE ABEND 10 22:12
00:01 #J17924 1
CHCKMRGE SUCC 10 22:12
00:01 #J17926 0
DATACLNS SUCC 10 22:12
00:01 #J17932 0
DATARMRG SUCC 10 22:13
00:01 #J18704 0
DBSELOAD SUCC 10 22:13
00:01 #J18706 7
DATAREPT SUCC 10 22:13
00:01 #J18712 0
DATARTRN SUCC 10 22:14
00:01 #J18714 0
$
```

O exemplo a seguir exibe o código de retorno para um job específico chamado`workstation#daily_db_load.dbseload`:

```
$ conman sj workstation#daily_db_load.dbseload\;keys\;retcod

Tivoli Workload Scheduler (UNIX)/CONMAN 8.4 (1.36.2.22) Materiais Licenciados -
Propriedade da IBM(R)
5698-WSH
(C) Copyright IBM Corp 1998, 2007 Todos os Direitos Reservados.
Direitos Restritos para Usuários do Governo dos Estados Unidos
Uso, duplicação ou divulgação restritos pelo documento
GSA ADP Schedule Contract com a IBM Corp.
IBM é uma marca registrada da International Business Machines
Corporation nos Estados Unidos e/ou em outros países.
Installed for user "tme10_99".
Locale LANG set to the following: "en"
Planejado para (Exp) 20/02/06 (#35) em CPUA. Batchman LIVES. Limit:50,Fence:0,Audit Level:0
Batchman ATIVO. Limit:50,Fence:0,Audit Level:0
sj workstation#daily_db_load.dbseload;keys;retcod 8
$
```

O recurso*retcod* quando integrado a um script pode tornar-se bem eficaz.

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Monitorar Tarefas**.
2. Selecione **Todas as Tarefas no plano** ou outra tarefa para monitorar tarefas.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.

Formato Padrão

CPU A estação de trabalho em que o job é executado.

Programar
O nome do fluxo de tarefa.

SchedTime
A hora e data em que a tarefa foi planejada para execução.

Tarefa O nome da tarefa. A seguinte notação pode preceder um nome de job:

>> rerun as
Um job que foi reexecutado com o comando **rerun**, ou como resultado de recuperação automática.

>> rerun step
Um job que foi reexecutado com o comando **rerun ;step**.

>> every run
A segunda e subseqüentes execuções de um job every.

>>recovery
A execução de um job de recuperação.

Estado
O estado do job ou fluxo de job. Os estados do job são os seguintes:

ABEND
O job terminou com um código de saída diferente de zero.

ABENP
Uma confirmação de **abend** foi recebida, mas o job não está concluído.

ADD A tarefa está sendo submetida.

CANCL
Apenas para dependências da interligação de redes. A tarefa remota ou o fluxo de tarefas foi cancelado.

DONE
A tarefa foi concluída em um estado desconhecido.

ERROR
Para dependências da interligação de redes apenas, ocorreu um erro durante a verificação do status remoto.

EXEC A tarefa está em execução.

EXTRN
O status é desconhecido apenas para as dependências da interligação de redes. Ocorreu um erro, uma ação reexecutar acabou de ser realizada na tarefa do fluxo de tarefas `EXTERNAL` ou a tarefa ou o fluxo de tarefas remoto não existe.

FAIL Impossível lançar a tarefa.

FENCE
A prioridade do job está abaixo da cerca.

HOLD
A tarefa está aguardando a resolução da dependência.

INTRO
O job é apresentado para ser lançado pelo sistema.

PEND A tarefa foi concluída e está aguardando confirmação.

READY
A tarefa está pronta para ser lançada e todas as dependências foram resolvidas.

SCHED
A hora **at** definida para o job não foi alcançada.

SUCC A tarefa foi concluída com um código de saída zero.

SUCCP
Uma confirmação SUCC foi recebida, mas a tarefa não foi concluída.

WAIT A tarefa está no estado WAIT (agente estendido).

Os estados dos fluxos de job são os seguintes:

ABEND
O fluxo de job terminou com um código de saída diferente de zero.

ADD O fluxo de tarefas foi adicionado com intervenção do operador.

CANCELP
O cancelamento do fluxo de tarefas está pendente. O cancelamento será adiado até que todas as dependências, incluindo uma hora at, sejam resolvidas.

ERROR
Para dependências da interligação de redes apenas, ocorreu um erro durante a verificação do status remoto.

EXEC O fluxo de tarefas está em execução.

EXTRN
Apenas para dependências da interligação de redes. Este é o estado do fluxo de tarefas EXTERNAL contendo tarefas que referenciam tarefas ou fluxos de tarefas na rede remota.

HOLD
O fluxo de tarefa está aguardando a resolução de dependências.

READY
O fluxo de job está pronto para ser lançado e todas as dependências foram resolvidas.

STUCK
A execução do fluxo de job foi interrompida. Nenhum job será lançado sem a intervenção do operador.

SUCC O fluxo de job foi concluído com êxito.

Pr A prioridade do fluxo de job ou job. Um sinal de mais (+) antes da prioridade significa que o job foi lançado.

(Est)Start
A hora de início do fluxo de job ou job. Parênteses indicam uma estimativa da hora de início. Se o comando for executado no mesmo dia em que a tarefa for planejada para execução, o parâmetro**Start** exibe um horário como (Est)Start. Se o comando for executado em um dia diferente do dia em que a tarefa for planejada para execução, o parâmetro**Start** exibe uma data como (Est)Start. Por exemplo, se você tiver a tarefa a seguir cujo horário de início ocorre no mesmo dia em que a tarefa está planejada para execução:

```
SCHEDULE MASTERB1#JS_B
ON RUNCYCLE RULE1 "FREQ=DAILY;"
AT 1700
:
MASTERB1#JOB1
 AT 1800
END
```

Você recebe a seguinte saída:

```
%sj @#@
                                          (Est) (Est)
CPU  Schedule SchedTime Job State Pr Start Elapse RetCode Deps
MASTERB1#JS_B  1700 08/18 ***** HOLD 10(17:00)
                          JOB1 HOLD 10(18:00)
```

Por exemplo, se você tiver a tarefa a seguir cujo horário de início ocorre em um dia diferente do dia em que a tarefa está planejada para execução:

```
SCHEDULE MASTERB1#JS_A
ON RUNCYCLE RULE1 "FREQ=DAILY;"
AT 0400
:
MASTERB1#JOB_A
 AT 0500
END
```

Você recebe a seguinte saída:

```
%sj @#@
                                              (Est) (Est)
CPU  Schedule SchedTime Job State Pr Start Elapse RetCode Deps
MASTERB1#JS_A  0400 08/19 ***** HOLD 10(08/19)
                              JOB_A HOLD 10(08/19)
```

(Est)Elapse
O tempo de execução do fluxo de job ou job. Parênteses indicam uma estimativa com base em estatísticas registradas.

dependências
Uma lista de dependências de job e comentários. Qualquer combinação das seguintes pode ser listada:

- Para uma dependência `follows`, é exibido um nome de fluxo de job ou de job.

  Se a tarefa ou o fluxo de tarefas for um predecessor pendente, seu nome será seguido por um `[P]`.

  No caso de uma dependência órfã, um `[O]` é exibido.

  Para obter informações adicionais sobre predecessores pendentes e dependências órfãs, consulte "Gerenciando Dependências de Acompanhamentos Externos para Tarefas e Fluxos de Tarefas" na página 61.
- Para uma dependência `opens`, é exibido o nome do arquivo. Se o arquivo residir em um agente estendido e seu nome tiver mais que 25 caracteres, apenas os últimos 25 serão exibidos.
- Para uma dependência `needs`, é exibido um nome de recurso colocado entre hifens (-). Se o número de unidades solicitadas for maior que um, o número será exibido antes do primeiro hífen.
- Para uma hora **deadline**, é exibida a hora antecedida por um sinal de menor (<).
- Para uma taxa **every**, é exibida a taxa de repetição antecedida de um e comercial (&).
- Para uma hora **until**, é exibida a hora antecedida por um sinal de menor (<).
- Para um tempo de **duração máxima** excedido, **[MaxDurationExceeded]** é exibido, além da configuração maxdur=*hhh:mm*.
- Para um tempo de **duração máxima** excedido, e para o qual o **`onmaxdur`** *action* está configurado como `Kill`, **[KillSubmitted]** é exibido.
- Para um tempo de **duração máxima** excedido, e para o qual o **`onmaxdur`** *action* está configurado como `Continue`, **[Continue]** será exibido.
- Para um tempo de **duração mínima** que não foi atingido e para o qual uma tarefa é concluída com êxito, [MinDurationNotReached] é exibido, além da configuração mindur=*hhh:mm*.
- Para um tempo de **duração mínima** que não foi atingido, e para o qual o **`onmindur`** *action* está configurado como `Continue`, **[Continue]** é exibido.
- Para um tempo de **duração mínima** que não foi atingido, e para o qual o **`onmindur`** *action* está configurado como `Abend`, **[Abended]** é exibido.

- Para um tempo de **duração mínima** que não foi atingido, e para o qual o **onmindur** *action* está configurado como `Confirm`, **[ConfirmSubmitted]** é exibido.
- Para uma dependência `prompt`, o número do prompt é exibido no formato #*num*. Para prompts globais, o nome do prompt segue entre parênteses.
- Para a execução de tarefas, o PID (Process Identification Number) é exibido no formato **#J***nnnnn*.
- As tarefas enviadas no UNIX utilizando os comandos Tivoli Workload Scheduler **at** e **batch** são etiquetadas como **[Userjcl]**.
- Ao relatar as dependências de hora, o comando **showjobs** mostra na coluna **Início**:
  - Somente a hora *hh:mm*, se o dia de configuração de dependências de tempo corresponde ao dia da execução do comando **showjobs**.
  - Somente a data *MM/DD* se o dia de configuração das dependências de hora não corresponder ao dia de execução do comando **showjobs**.
- Jobs cancelados são rotulados como **[Cancelled]**.
- Jobs cancelados com a opção **;pend** são rotulados como **[Cancel Pend]**.
- Jobs com horas **until** expiradas, incluindo jobs cancelados com a opção **;pend** são rotulados como **[Until]**.
- **[Recovery]** significa que a operação requer intervenção.
- **[Confirmed]** significa que confirmação é necessária, pois a foi planejada usando a palavra-chave **confirm**.
- **[Script]** aplica-se apenas a redes de ponta a ponta; isso significa que esse job tem um script centralizado e que o IBM Tivoli Workload Scheduler for z/OS ainda não o transferiu por download para o agente.

Formato Keys

Os nomes de job são listados um em cada linha no seguinte formato:

```
workstation#jstream hhmm mm/dd.job
```

por exemplo:

```
CPU  Schedule SchedTime Job State Pr Start Elapse RetCode Deps

MYCPU+#SCHED_F+ 0600 03/04 ******* HOLD 55(03/04)    [03/04/06]; #33
                (M235062+#)JOBMDM HOLD 30(03/04)    #1(PRMT3);-16 JOBSLOTS-

MYCPU+#SCHED_F+ 1010 03/04 ******* HOLD 55(03/04)    [03/04/06]; #34
                (M235062+#)JOBMDM HOLD 30(03/04)    #1(PRMT3);-16 JOBSLOTS-
```

Formato Info

CPU A estação de trabalho em que o job é executado.

Programar
O nome do fluxo de tarefa.

SchedTime
A hora e data em que a tarefa foi planejada para execução.

Tarefa O nome da tarefa. A seguinte notação pode preceder um nome de job:

>> rerun as
Um job que foi reexecutado com o comando **rerun**, ou como resultado de recuperação automática.

>> rerun step
Um job que foi reexecutado com o comando **rerun ;step**.

>> every run
A segunda e subseqüentes execuções de um job every.

>>recovery
A execução de um job de recuperação.

Arquivo de Tarefa
O nome do script ou do arquivo executável do job. Nomes longos de arquivos podem causar reinício cíclico, causando paginação incorreta. Para evitar isso, direcione a saída para **more**.

Opt A opção de recuperação do job, se houver. As opções de recuperação são **RE** para reexecução, **CO** para continuar e **ST** para parar.

Tarefa O nome do job de recuperação, se houver.

Aviso O número do prompt de recuperação, se houver.

Por exemplo:

```
conman "sj;info | more
```

produz uma saída de amostra como a seguinte:

```
 --------Restart---------
CPU  Schedule SchedTime Job  JobFile    Opt Job Prompt
M235062+#SCHED_22 1010 03/06
                    JOBMDM /usr/acct/scripts/gl1
                  (B236153+#)JOB_FTA echo job12
M235062+#SCHED_22 0600 03/07
                    JOBMDM /usr/acct/scripts/gl1
                  (B236153+#)JOB_FTA echo job12
M235062+#FINAL 2359 02/13
                    STARAPPSERVER /opt/IBM/TWA/TWS/../wastools/startWas.sh
                                                                  CO
                    MAKEPLAN /opt/IBM/TWA/TWS/MakePlan TWSRCMAP:(RC=0) OR (RC=4)
                    SWITCHPLAN /opt/IBM/TWA/TWS/SwitchPlan
M235062+#FINALPOSTREPORTS 2359 02/13
                    CHECKSYNC /opt/IBM/TWA/TWS/CheckSync
                    CREATEPOSTREPORTS /opt/IBM/TWA/TWS/CreatePostReports
                                                                  CO
                    UPDATESTATS /opt/IBM/TWA/TWS/UpdateStats
                                                                  CO
M235062+#SCHED12 1010 03/06
                    JOBMDM /usr/acct/scripts/gl1
                  (B236153+#)JOB_FTA echo job12
```

Formato Step

Esse formato não é suportado no Windows.

CPU A estação de trabalho em que o job é executado.

Programar
O nome do fluxo de tarefa.

SchedTime
A hora e data em que a tarefa foi planejada para execução.

Tarefa O nome da tarefa. A seguinte notação pode preceder um nome de job:

>> rerun as
Um job que foi reexecutado com o comando **rerun**, ou como resultado de recuperação automática.

>> repeated as
A segunda e subseqüentes execuções de um job **every**.

Estado
O estado do job ou fluxo de job. Consulte “Formato Padrão” para obter informações sobre estado.

Código de retorno
O código de retorno da tarefa.

Job# O número de identificação do processo exibido como **#J***nnnnn*.

Etapa Uma lista de processos descendentes associados ao job. Para jobs do agente estendido, são listados apenas os processos do host.

Formato Logon

CPU A estação de trabalho em que o job é executado.

Programar
O nome do fluxo de tarefa.

SchedTime
A hora e data em que a tarefa foi planejada para execução.

Tarefa O nome da tarefa. A seguinte notação pode preceder um nome de job:

>> rerun as
Um job que foi reexecutado com o comando **rerun**, ou como resultado de recuperação automática.

>> repeated as
A segunda e subseqüentes execuções de um job **every**.

Estado
O estado do job ou fluxo de job. Consulte “Formato Padrão” para obter informações sobre estado.

Código de retorno
O código de retorno da tarefa.

Job# O número de identificação do processo exibido como **#J***nnnnn*.

Logon O nome do usuário sob o qual o job é executado.

Em sistemas operacionais Windows, é possível ter um dos seguintes formatos:

nome do usuário
Em que *username* é o nome do usuário do Windows.

domain\username
Em que *domain* é o domínio do Windows do usuário e o *username* é o nome do usuário do Windows.

username@internet_domain
Em que *username@internet_domain* é o nome de um usuário do sistema em um formato de endereço de e-mail. O nome do usuário é seguido pelo "sinal" seguido pelo nome do domínio da Internet com o qual o usuário está associado.

Nota: Insira o caractere de escape '\' antes do caractere '@' no valor `username@internet_domain` no campo de logon. Por exemplo, se você estiver usando o usuário `administrator@bvt.com` no campo de logon, utilize a seguinte sintaxe:

..... ; logon=administrator\@bvt.com

Formato Stdlist

Um arquivo de lista padrão é criado automaticamente pelo **jobmon** no Windows ou no **jobman** no UNIX, para cada job que o **jobmon** e o**jobman** ativar. É possível exibir o conteúdo dos arquivos de lista padrão utilizando o **conman**. Um arquivo de lista padrão contém:

- Banners de cabeçalho e de encerramento.
- Comandos de eco.
- A saída stdout do job.
- A saída stderr do job.

Para especificar um determinado formato de data a ser utilizado nos arquivos de lista padrão, altere o formato de data do IBM Tivoli Workload Scheduler antes de criar os arquivos de lista padrão. Isso é feito modificando-se o formato do código de idioma da data.

Dependendo do ambiente, altere o formato do código de idioma da data executando as etapas listadas a seguir:

- No UNIX, defina a variável LANG no ambiente quando o **netman** for iniciado. Se a variável LANG não for definida, o código de idioma do sistema operacional será definido, por padrão, como "C".
- No Windows, execute as etapas a seguir:
  1. Vá para Painel de Controle→Opções Regionais e defina seu código de idioma (local.
  2. Clique com o botão direito do mouse em **Meu Computador**, vá para Propriedades, clique em **Avançado**, vá para Variáveis de Ambiente e defina a variável LANG como uma variável do sistema.
  3. Encerre e reinicie o sistema.

São exibidos os arquivos de lista padrão dos jobs selecionados.

Formato Stdlist;keys

Os nomes dos arquivos de lista padrão dos jobs selecionados são listados, um em cada linha.

Formato Crit

CPU A estação de trabalho em que o job é executado.

Programar
O nome do fluxo de tarefa.

SchedTime
A hora e data em que a tarefa foi planejada para execução.

Tarefa O nome da tarefa. A seguinte notação pode preceder um nome de job:

>> rerun as
Um job que foi reexecutado com o comando **rerun**, ou como resultado de recuperação automática.

>> repeated as
A segunda e subseqüentes execuções de um job **every**.

Estado
O estado do job ou fluxo de job. Consulte “Formato Padrão” para obter informações sobre estado.

Pr A prioridade do fluxo de job ou job. Um sinal de mais (+) antes da prioridade significa que o job foi lançado.

(Est)Start
A hora de início do fluxo de job ou job. Parênteses indicam uma estimativa da hora de início. Se a hora de início ultrapassar 24 horas anteriores ou posteriores, a data será listada no lugar da hora.

(Est)Elapse
O tempo de execução do fluxo de job ou job. Parênteses indicam uma estimativa com base em estatísticas registradas.

CP Indica se a tarefa está sinalizada como crítica (**C**) e/ou promovida (**P**).

CritStart
A hora mais recente que a tarefa pode ser iniciada sem influenciar nos prazos finais de sucessores imprescindíveis.

Por exemplo, o resultado do seguinte comando genérico:

```
%sj @#@;crit
```

é:

```
                                          (Est) (Est)  Crit
CPU  Schedule SchedTime Job  State Pr Start Elapse CP Start

MYCPU_F+#JSA  1600 03/05 ******** HOLD 10
                         JOBA1 HOLD 10    CP 1759 03/05
                         JOBA2 HOLD 10      1758 03/05
                         JOBA3 HOLD 10      1757 03/05
                         JOBA4 HOLD 10    C 1659 03/05
```

Observe que :

- O sinalizador **C** aplica-se a todas as tarefas definidas como críticas na definição de fluxo de tarefas. É configurado no plano ou horário de **submit**.
- O sinalizador **P** aplica-se a tarefas críticas e a seus predecessores (que são tarefas que não são definidas como críticas, mas que não devem impactar na conclusão em tempo hábil de uma tarefa crítica do sucessor). Ele é configurado em tempo de execução se a tarefa foi promovida.
- Tanto as tarefas críticas quanto os predecessores críticos têm um horário de início crítico.

  O planejador calcula o horário de início crítico de uma tarefa crítica subtraindo sua duração estimada do respectivo prazo final. O planejador calcula o horário de início crítico de um predecessor crítico subtraindo sua duração estimada do horário de início crítico de seu próximo sucessor. Em uma rede crítica, o planejador calcula o horário de início crítico da tarefa crítica primeiro e, em seguida, trabalha de modo inverso ao longo da cadeia de predecessores. Esses cálculos são reiterados quantas vezes forem necessárias até a execução da tarefa crítica.

Formato Deps

Jobs utilizados em dependências `follows` são listados, seguidos pelos jobs e fluxos de job dependentes. Os jobs são listados no formato **showjobs.** Os fluxos de job são listados no formato **showschedules** padrão.

Formato Deps;keys

Os jobs e fluxos de job que possuem dependências `follows` são listados, um em cada linha.

Formato Deps;info

Os jobs utilizados em dependências `follows` são listados, seguidos pelos jobs e fluxos de job dependentes. Os jobs são listados no formato **showjobs;info.** Os fluxos de job são listados no formato **showschedules** padrão.

Formato Deps;logon

Os jobs utilizados em dependências `follows` são listados, seguidos pelos jobs e fluxos de job dependentes. Os jobs são listados no formato **showjobs;logon**. Os fluxos de job são listados no formato **showschedules** padrão.

showprompts

Exibe informações sobre prompts.

As informações exibidas são atualizadas somente se oTivoli Workload Scheduler (batchman) estiver em execução. A confirmação da atividade ou inatividade do batchman aparece na tela através da mensagem `Batchman LIVES` ou `Batchman down` quando você emite o comando `conman start`.

Você deverá ter acesso ***list*** para o objeto que está sendo mostrado se a opção ***enListSecChk*** tiver sido configurada como **yes** no gerenciador de domínio principal quando o plano de produção foi criado ou estendido.

Sintaxe

{**showprompts** | **sp**} [*promptselect*]
[**;keys**]
[**;offline**]

{**showprompts** | **sp**} [*promptselect*]
[**;deps**[**;keys** | **info** | **logon**]][**;offline**]

Argumentos

promptselect
[*promptname* | [*workstation*#]*msgnum*][**;***state*[**;**...]]

promptname
Especifica o nome de um prompt global. Caracteres curinga são permitidos.

estação de trabalho
Especifica o nome da estação de trabalho na qual um prompt não nomeado foi emitido. O padrão é a estação de trabalho na qual o **conman** está sendo executado.

msgnum
Especifica o número de mensagem de um prompt não nomeado.

state Especifica o estado dos prompts a serem exibidas. Os estados são os seguintes:

SIM O prompt recebeu a resposta **y**.

NÃO O prompt recebeu a resposta **n**.

PERGUNTADO
O prompt foi emitido, mas não recebeu resposta.

INACT
O prompt não foi emitido.

keys Exibe uma única coluna de lista dos objetos selecionados pelo comando.

deps Exibe informações no formato **deps**. Utilize **keys**, **info** ou **logon** para modificar a exibição.

info Exibe informações no formato **info**.

logon Exibe informações no formato **logon**.

off-line
Envia a saída do comando para o dispositivo de saída do **conman**. Para obter informações sobre esse dispositivo, consulte "Saída Off-line" na página 340.

Nota: Números de prompt designados para prompts globais e locais são alterados quando o plano de produção é estendido.

Resultados

A saída do comando é produzida em três formatos: **padrão**, **keys** e **deps**. Os argumentos **keys**, **info** e **logon** modificam a exibição de **deps**.

Exemplos

Para exibir o status de todos os prompts emitidos na estação de trabalho na qual o **conman** está sendo executado, execute o seguinte comando:

```
showprompts
```

segue uma amostra:

```
State Message or Prompt
ASKED 1(PRMT3) !continue?
INACT 3(CPUA#SCHED_12[(0600 03/12/06),(0AAAAAAAAAAABST)]) Are you ready to process job1?
      INACT 3(CPUA#SCHED_12[(0600 03/12/06),(0AAAAAAAAAAABST)]) Are you ready to process job1?
INACT 5(CPUA#SCHED_12[(1010 03/12/06),(0AAAAAAAAAAABSU)]) Are you ready to process job2?
      INACT 3(CPUA#SCHED_12[(0600 03/12/06),(0AAAAAAAAAAABST)]) Are you ready to process job1?
INACT 7(CPUA#SCHED_22[(0600 03/12/06),(0AAAAAAAAAAABTR)]) Are you ready to process job3?
      INACT 3(CPUA#SCHED_12[(0600 03/12/06),(0AAAAAAAAAAABST)]) Are you ready to process job1?
```

Para exibir o status de todos os mis prompts emitidos no formato **deps**, execute o seguinte comando:

```
sp mis@;asked;deps
```

Para exibir o status do prompt número 7 na estação de trabalhoCPUA, execute o seguinte comando:

```
sp CPUA#7
```

A saída do comando é:

```
INACT 7(CPUA#SCHED_22[(0600 03/12/06),(0AAAAAAAAAAABTR)]) Are you ready to process job3?
      INACT 3(CPUA#SCHED_12[(0600 03/12/06),(0AAAAAAAAAAABST)]) Are you ready to process job1?
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Prompts do Monitor**.
2. Selecione **Todos os Prompts no Plano** ou outra tarefa para monitorar prompts.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.

Formato Padrão

Estado
O estado do prompt.

Mensagem ou Prompt
Para prompts nomeados, o número da mensagem, o nome e o texto do prompt. Para prompts não nomeados, o número da mensagem, o nome do job ou fluxo de job e o texto do prompt.

Formato Keys

Os prompts são listados um em cada linha. Prompts nomeados são listados com seus nomes e números de mensagens. Prompts não nomeados são listados com seus números de mensagens e os nomes dos jobs ou fluxos de job nos quais aparecem como dependências.

Formato Deps

Os prompts utilizados como dependências são listados, seguidos pelos jobs e fluxos de job dependentes. Os jobs são listados no formato **showjobs.** Os fluxos de job são listados no formato **showschedules** padrão.

Formato Deps;keys

Os jobs e os fluxos de job que possuem dependências `prompt` são listados um em cada linha.

Formato Deps;info

Os prompts utilizados como dependências são listados, seguidos pelos jobs e fluxos de job dependentes. Os jobs são listados no formato **showjobs;info.** Os fluxos de job são listados no formato **showschedules** padrão.

Formato Deps;logon

Os prompts utilizados como dependências são listados, seguidos pelos jobs e fluxos de job dependentes. Os jobs são listados no formato **showjobs;logon**. Os fluxos de job são listados no formato **showschedules** padrão.

showresources

Exibe informações sobre recursos.

As informações exibidas são atualizadas somente se oTivoli Workload Scheduler (batchman) estiver em execução. A confirmação da atividade ou inatividade do batchman aparece na tela através da mensagem `Batchman LIVES` ou `Batchman down` quando você emite o comando `conman start`.

Você deverá ter acesso ***list*** para o objeto que está sendo mostrado se a opção ***enListSecChk*** tiver sido configurada como **yes** no gerenciador de domínio principal quando o plano de produção foi criado ou estendido.

Sintaxe

{**showresources** | **sr**} [[*workstation***#**]*resourcename*]
[**;keys**]
[**;offline**]

{**showresources** | **sr**} [[*workstation***#**]*resourcename*]
[**;deps**[**;keys** | **info** | **logon**]]
[**;offline**]

Argumentos

estação de trabalho
Especifica o nome da estação de trabalho na qual o recurso foi definido. O padrão é a estação de trabalho na qual o **conman** está sendo executado.

resourcename
Especifica o nome do recurso. Caracteres curinga são permitidos.

keys Exibe uma única coluna de lista dos objetos selecionados pelo comando.

deps Exibe informações no formato **deps**. Utilize **keys**, **info** ou **logon** para modificar a exibição.

info Exibe informações no formato **info**.

logon Exibe informações no formato **logon**.

off-line
Envia a saída do comando para o dispositivo de saída do **conman**. Para obter informações sobre esse dispositivo, consulte "Saída Off-line" na página 340.

Resultados

A saída do comando é produzida em três formatos: **padrão**, **keys** e **deps**. Os argumentos **keys**, **info** e **logon** modificam a exibição de **deps**.

Exemplos

Para exibir informações sobre todos os recursos na estação de trabalho na qual o **conman** está sendo executado, execute o seguinte comando:

```
showresources
```

Uma saída de amostra é:

```
CPU#Resource  Total Available Qty UsedBy
CPUA #JOBSLOTS 16  16  No holders of this resource
```

Para exibir informações sobre o recurso`jobslots` na estação de trabalho`CPUA` no formato **deps**, execute o seguinte comando:

```
sr CPUA#JOBSLOTS;deps
```

Uma saída de amostra é a seguinte:

```
                                                (Est) (Est)
Workstation Job Stream SchedTime Job State Pr Start Elapse RetCode Dependencies

CPUA  #JOBSLOTS Dependencies are:


  FTAA  #SCHED_F+ 0600 03/04 ****** HOLD 55(03/04)    [03/04/06];#33
                        (CPUA#)JOBMDM HOLD 30(03/04)    #1(PRMT3);-16 JOBSLOTS-

  FTAA  #SCHED_F+ 1010 03/04 ****** HOLD 55(03/04)    [03/04/06];#34
                        (CPUA#)JOBMDM HOLD 30(03/04)    #1(PRMT3);-16 JOBSLOTS-

  FTAA  #SCHED_F+ 0600 03/05 ****** HOLD 55(03/05)    [03/04/06];#35
                        (CPUA#)JOBMDM HOLD 30(03/05)    #1(PRMT3);-16 JOBSLOTS-
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Prompts do Monitor**.
2. Selecione **Todos os Recursos no Plano** ou outra tarefa para monitorar recursos.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.

Formato Padrão

CPU A estação de trabalho na qual o recurso foi definido.

Resource
O nome do recurso.

Total O número total de unidades de recurso definidas.

Disponível
O número de unidades de recurso que não foram alocadas.

Qty O número de unidades de recurso alocadas a um job ou fluxo de job.

Used By
O nome do job ou fluxo de job.

Formato Keys

Os recursos são listados um em cada linha.

Formato Deps

Os recursos utilizados como dependências são listados, seguidos pelos jobs e fluxos de job dependentes. Os jobs são listados no formato **showjobs.** Os fluxos de job são listados no formato **showschedules** padrão.

Formato Deps;keys

Os jobs e os fluxos de job que possuem dependências `resource` são listados um em cada linha.

Formato Deps;info

Os recursos utilizados como dependências são listados, seguidos pelos jobs e fluxos de job dependentes. Os jobs são listados no formato **showjobs;info.** Os fluxos de job são listados no formato **showschedules** padrão.

Formato Deps;logon

Os recursos utilizados como dependências são listados, seguidos pelos jobs e fluxos de job dependentes. Os jobs são listados no formato **showjobs;logon**. Os fluxos de job são listados no formato **showschedules** padrão.

showschedules

Exibe informações sobre fluxos de job.

As informações exibidas são atualizadas somente se oTivoli Workload Scheduler (batchman) estiver em execução. A confirmação da atividade ou inatividade do batchman aparece na tela através da mensagem `Batchman LIVES` ou `Batchman down` quando você emite o comando `conman start`.

Você deverá ter acesso ***list*** para o objeto que está sendo mostrado se a opção ***enListSecChk*** tiver sido configurada como **yes** no gerenciador de domínio principal quando o plano de produção foi criado ou estendido.

Sintaxe

{**showscheds** | **ss**} [*jstreamselect*]
[;keys]
[;offline]
[;showid]

{**showscheds** | **ss**} [*jstreamselect*]
[;deps[;keys | info | logon]]
[;offline]
[;showid]

Argumentos

jstreamselect
Consulte "Selecionando Fluxos de Tarefas em Comandos" na página 353.

keys Exibe uma única coluna de lista dos objetos selecionados pelo comando.

deps Exibe informações no formato deps; ou seja, os fluxos de tarefas usados em dependências de *sequências* são listados seguidos pelas tarefas e fluxos de tarefas dependentes. As tarefas são listadas no formato `showjobs` básico. Os fluxos de tarefas são listados no formato `showschedules` básico. Use "keys", "info" ou "logon" para modificar a exibição de "deps".

info Exibe informações no formato **info**.

logon Exibe informações no formato **logon**.

off-line
Envia a saída do comando para o dispositivo de saída do **conman**. Para obter informações sobre esse dispositivo, consulte "Saída Off-line" na página 340.

showid
Exibe para cada fluxo de tarefa o identificador do fluxo de tarefa.

Resultados

A saída do comando é produzida em três formatos: padrão, **keys** e **deps**. Os argumentos **keys**, **info** e **logon** modificam a exibição de **deps**. A lista exibida na saída do comando não inclui tarefas que foram executadas novamente nos processos de planejamento anteriores, mas o total mostrado no final inclui.

Exemplos

Para exibir o status do fluxo de tarefa `CLEM_DOCOM` na estação de trabalho`site3` e solicitar o identificador do fluxo de tarefa, execute o seguinte comando:

```
%ss @#JS_DOCOM ;showid
```

Uma saída de amostra desse comando é a seguinte:

```
                                    (Est) (Est)  Jobs Sch
Workstation Job Stream  SchedTime State Pr Start Elapse # OK Lim
site3  #JS_DOCOM 0600 11/26 SUCC 10 11/26 00:01 1 1  {0AAAAAAAAAAAACRZ}
```

Para exibir o status de todos os fluxos de job no estado HOLD na estação de trabalho na qual o **conman** está sendo executado, execute o seguinte comando:

```
showschedules @+state=hold
```

Uma saída de amostra para este comando é a seguinte:

```
                                          (Est) (Est)  Jobs Sch
Workstation Job Stream  SchedTime State Pr Start Elapse # OK Lim
site3  #FILE_JS1 0600 11/26 HOLD 10 (11/26)   1 0  parms FILE_JS1`
```

Para exibir o status de todos os fluxos de tarefas com o nome iniciado por sched na estação de trabalhoCPUA no formato **deps;info**, execute o seguinte comando:

```
ss CPUA#sched@;deps;info
```

Uma saída de amostra é a seguinte:

```
    --------Restart---------
  CPU  Schedule SchedTime Job  JobFile    Opt Job Prompt

CPUA #JS_FIRST1[(0600 03/10/06),(0AAAAAAAAAAAABVY)] Dependencies are:
  CPUA#MOD 0212 03/10
                         JOBMDM /usr/scripts/gl1(B236153+#)JOB_FTA1 echo Start gl1?
  CPUA#MOD 0251 03/10
                         JOBMDM /usr/scripts/gl2(B236153+#)JOB_FTA2 echo Start gl2?
```

Para exibir o status **off-line** de todos os fluxos de job no estado ABEND em todas as estações de trabalho, execute o seguinte comando:

```
ss @#@+state=abend;off
```

Para exibir o status de todos os fluxos de job em todas as estações de trabalho, execute o seguinte comando:

```
%ss @#@
```

Esta é uma saída de amostra para o comando:

```
                                          (Est) (Est)  Jobs Sch
Workstation Job Stream  SchedTime State Pr Start Elapse # OK Lim
site3  #JS_DOCOM 0600 11/26 SUCC 10 11/26 00:01 1 1
site3  #JS_SCRIPT 0600 11/26 SUCC 10 11/26 00:03 1 1
site2  #JS_PRED1 1000 11/26 SUCC 10 11/26 00:01 1 1
site3  #JS_SCRIPT1 0600 11/26 ABEND 10 11/26 00:01 1 0
site3  #LFILEJOB  0600 11/26 READY 10     1 0
site1  #RES_100  0600 11/26 SUCC 10 11/26 00:09 1 1
site3  #FILE_JS1 0600 11/26 HOLD 10 (11/26)   1 0  parms FILE_JS1`
site3  #FILE_JOB 0600 11/26 SUCC 10 11/26 00:01 1 1
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Carga de Trabalho** > **Monitorar Fluxos de Tarefas**.
2. Selecione **Todos os Fluxos de Tarefas no plano** ou outra tarefa para monitorar fluxos de tarefas.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.

Formato Padrão

CPU A estação de trabalho na qual o fluxo de job é executado.

Programar
O nome do fluxo de tarefa.

SchedTime
A hora e data em que o fluxo de tarefas foi planejado para execução.

Estado
O estado do fluxo de job. Os estados são os seguintes:

ADD O fluxo de tarefas foi adicionado com intervenção do operador.

ABEND
O fluxo de job terminou com um código de saída diferente de zero.

CANCELP
O cancelamento do fluxo de tarefas está pendente. O cancelamento será adiado até que todas as dependências, incluindo uma hora at, sejam resolvidas.

ERROR
Para dependências da interligação de redes apenas, ocorreu um erro durante a verificação do status remoto.

EXEC O fluxo de tarefas está em execução.

EXTRN
Apenas para dependências da interligação de redes. Este é o estado do fluxo de tarefas EXTERNAL contendo tarefas que referenciam tarefas ou fluxos de tarefas na rede remota.

HOLD
O fluxo de job está aguardando a resolução da dependência.

READY
O fluxo de job está pronto para ser lançado e todas as dependências foram resolvidas.

STUCK
A execução do fluxo de job foi interrompida. Nenhum job será lançado sem a intervenção do operador.

SUCC O fluxo de job foi concluído com êxito.

Pr A prioridade do fluxo de tarefas.

(Est)Start
A hora de início do fluxo de job ou job. Parênteses indicam uma estimativa da hora de início. Se o comando for executado no mesmo dia em que o fluxo de tarefas for planejado para execução, o parâmetro**Start** exibe um horário como (Est)Start. Se o comando for executado em um dia diferente do dia em que o fluxo de tarefas for planejado para execução, o parâmetro Start exibe uma data como (Est)Start. Por exemplo, se você tiver o fluxo de tarefas a seguir cujo horário de início ocorre no mesmo dia em que o fluxo de tarefas está planejado para execução:

```
SCHEDULE MASTERB1#JS_B
ON RUNCYCLE RULE1 "FREQ=DAILY;"
AT 1800
:
MASTERB1#JOB1
END
```

Você recebe a seguinte saída:

```
%ss 0#0
                                        (Est) (Est)  Jobs Sch
CPU  Schedule SchedTime State Pr Start Elapse # OK Lim
MASTERB1#JS_B  1800 08/18 HOLD 10(18:00)   1 0
```

Por exemplo, se você tiver o fluxo de tarefas a seguir cujo horário de início ocorre em um dia diferente do dia em que o fluxo de tarefas está planejado para execução:

```
SCHEDULE MASTERB1#JS_A
ON RUNCYCLE RULE1 "FREQ=DAILY;"
AT 0500
:
MASTERB1#JOB1
END
```

Você recebe a seguinte saída:

```
%ss @#@
                                        (Est) (Est)  Jobs Sch
CPU  Schedule SchedTime State Pr Start Elapse # OK Lim
MASTERB1#JS_A  0500 08/19 HOLD 10(08/19)  1 0
```

(Est)Elapse
O tempo de execução do fluxo de job. Parênteses indicam uma estimativa com base em estatísticas registradas.

Jobs # O número de jobs do fluxo de job.

Jobs OK
O número de jobs que foram concluídos com êxito.

Sch Lim
O limite de job do fluxo de job. Se não houver um listado, o limite não estará em vigor.

dependências
Uma lista de dependências de fluxo de job e comentários. Qualquer combinação das seguintes pode ser listada:

- Para uma dependência `follows`, é exibido um nome de fluxo de job ou de job. Se a tarefa ou o fluxo de tarefas for um predecessor pendente, seu nome será seguido por um `[P]`.
- Para uma dependência `opens`, é exibido o nome do arquivo. Se o arquivo residir em um agente estendido e seu nome tiver mais que 25 caracteres, apenas os últimos 25 serão exibidos.
- Para uma dependência `needs`, é exibido um nome de recurso colocado entre hifens (-). Se o número de unidades solicitadas for maior que um, o número será exibido antes do primeiro hífen.
- Para uma hora **until**, a hora antecedida por um sinal de menor (<).
- Para uma dependência `prompt`, o número do prompt exibido como #*num*. Para prompts globais, o nome do prompt segue entre parênteses.
- Fluxos de job cancelados são rotulados como **[Cancelled]**.
- Fluxos de job cancelados com a opção **;pend** são rotulados como **[Cancel Pend]**.
- Para uma hora **deadline**, é exibida a hora antecedida por um sinal de menor (<).
- Fluxos de job que contêm a palavra-chave **carryforward** são rotulados como **[Carry]**.
- Para fluxos de tarefas que foram transportados do plano de produção anterior, o nome e data originais são exibidos entre colchetes.
- Ao relatar as dependências de hora, o comando **showschedules** mostra na coluna **Início**:
  - Somente a hora *hh:mm*, se o dia de configuração de dependências de tempo corresponde ao dia da execução do comando **showschedules**.
  - Somente a data *mm/dd* se o dia de configuração das dependências de hora não corresponder ao dia de execução do comando **showschedules**.

  Nota: A hora ou data exibida na coluna **Início** é convertida para o fuso horário configurado na estação de trabalho onde o fluxo de tarefa deve ser executado.

Formato Keys

Os fluxos de job são listados um em cada linha.

Formato Deps

Os fluxos de job utilizados como dependências são listados, seguidos pelos jobs e fluxos de job dependentes. Os jobs são listados no formato **showjobs.** Os fluxos de job são listados no formato **showschedules** padrão.

Formato Deps;keys

Os fluxos de job que possuem dependências `follows` são listados, um em cada linha.

Formato Deps;info

Os fluxos de job utilizados como dependências são listados, seguidos pelos jobs e fluxos de job dependentes. Os jobs são listados no formato **showjobs;info.** Os fluxos de job são listados no formato **showschedules** padrão.

Formato Deps;logon

Os fluxos de job utilizados como dependências são listados, seguidos pelos jobs e fluxos de job dependentes. Os jobs são listados no formato **showjobs;logon**. Os fluxos de job são listados no formato **showschedules** padrão.

shutdown

Para de forma incondicional todos os processos e serviços de produção do Tivoli Workload Scheduler, inclusive **batchman**, **jobman**, **netman**, **mailman**, **appservman**, todos os servidores **mailman** e todos os processos **writer**.

Apesar de esse comando parar o serviço**appservman**, ele não para os serviços do WebSphere Application Server. Para parar serviços do WebSphere Application Server, execute o comando **stopappserver**. Para obter informações adicionais, consulte a "stopappserver" na página 445.

Nas estações de trabalho do Windows, o comando **shutdown** não para o serviço**tokensrv**.

Nota: Esse comando não é suportado em estações de trabalho de mecanismo remoto.

Você deve ter acesso ***shutdown*** à estação de trabalho.

Sintaxe

{**shutdown** | **shut**} [**;wait**]

Argumentos

wait Aguarda até que todos os processos tenham parado antes de solicitar outro comando.

Comentários

O comando **shutdown** pára os processos somente na estação de trabalho em que o **conman** está sendo executado. Para reiniciar apenas o**netman**, execute o comando **StartUp**. Para obter informações sobre o comando **StartUp**, consulte "StartUp" na página 527. Para reiniciar a árvore do processo inteira, execute os seguintes comandos **conman**:

```
start
startappserver
startmon
```

Você deve executar um comando **unlink @** do **conman** antes de executar um comando **shutdown**.

Exemplos

Para encerrar a produção na estação de trabalho na qual o **conman** está sendo executado, execute o seguinte comando:

```
unlink @
shutdown
```

Para encerrar a produção na estação de trabalho na qual o **conman** está sendo executado e aguardar a parada de todos os processos, execute o seguinte comando:

```
unlink@;noask
shut ;wait
```

start

Inicia os processos de produção do Tivoli Workload Scheduler, exceto o mecanismo de monitoramento de eventos e oWebSphere Application Server (consulte "startappserver" na página 439 e "startmon" na página 441 para aprender sobre os comandos que iniciam esses processos).

Nota: Certifique-se de que **conman start** não seja emitido enquanto **JnextPlan** ou **stageman** for executado.

Você deve ter acesso ***start*** para a estação de trabalho.

Sintaxe

start [*domain***!**]*workstation*
[;mgr]
[;noask]
[;demgr]

Argumentos

domínio
Especifica o nome do domínio no qual as estações de trabalho foram iniciadas. Caracteres curinga são permitidos.

Este argumento é útil quando se está iniciando mais de uma estação de trabalho em um domínio. Por exemplo, para iniciar todos os agentes no domínio**stlouis**, utilize o seguinte comando:

```
start stlouis!@
```

Se *domain* for omitido e *workstation* contiver caracteres curinga, o domínio padrão será aquele no qual o **conman** está sendo executado.

estação de trabalho
Especifica o nome da estação de trabalho a ser iniciada. Caracteres curinga são permitidos.

Esse comando não é suportado em estações de trabalho de mecanismo remoto.

mgr Pode ser digitado apenas na estação de trabalho na qual o **conman** está sendo executado. Ele inicia a estação de trabalho local como o gerenciador de domínio. A estação de trabalho torna-se o novo gerenciador de domínio e o gerenciador de domínio atual torna-se um agente tolerante a falhas. Essa forma do comando segue, geralmente, um comando **stop**.

Nota: O método preferido de se comutar um gerenciador de domínio é utilizar um comando **switchmgr**. Consulte "switchmgr" na página 464 para obter informações adicionais.

noask Especifica para não solicitar confirmação antes da tomada de ação em cada estação de trabalho qualificada.

demgr A opção evita a abertura de conexões externas durante o tempo de transição quando o agente é iniciado como um gerenciador de domínio antigo e quando o comando **switchmgr** é executado, impedindo a função do agente do gerenciador de domínio. Essa opção é executada automaticamente, mas até que o gerenciador de domínio antigo tenha processado o evento**switchmgr** (no caso, por exemplo, do reinício retardado ou do reinício após o reparo de um agente danificado), a opção *demgr* **deverá** ser utilizada para iniciar o gerenciador de domínio antigo a partir da linha de comandos local. Para obter detalhes adicionais sobre esta opção, consulte o Guia de Administração do *Tivoli Workload Scheduler*.

Comentários

O comando **start** é utilizado no início de cada período de produção para reiniciar o Tivoli Workload Scheduler depois do processamento de pré-produção. Nessa hora, faz com que os agente tolerante a falhas e os agentes padrão vinculados automaticamente sejam inicializados e iniciados automaticamente. Os agentes que não são vinculados automaticamente são inicializados e iniciados quando você executa um comando **link**.

Supondo-se que um usuário tenha acesso ***start*** a estações de trabalho sendo iniciadas, as seguintes regras serão aplicadas:

- Um usuário executando o**conman** no gerenciador de domínio principal pode iniciar qualquer estação de trabalho na rede.
- Um usuário executando o**conman** em um gerenciador de domínio diferente do principal pode iniciar qualquer estação de trabalho nesse domínio e em domínios subordinados. O usuário não pode iniciar estações de trabalho em domínios ponto a ponto.
- Um usuário executando o**conman** em um agente pode iniciar estações de trabalho hospedadas por esse agente.

Exemplos

A Figura 24 na página 439 e a Tabela 68 na página 439 abaixo mostram as estações de trabalho iniciadas pelos comandos **start** executados por usuários em vários locais da rede.

DM*n* são gerenciadores de domínio e **A***nn* são agentes.

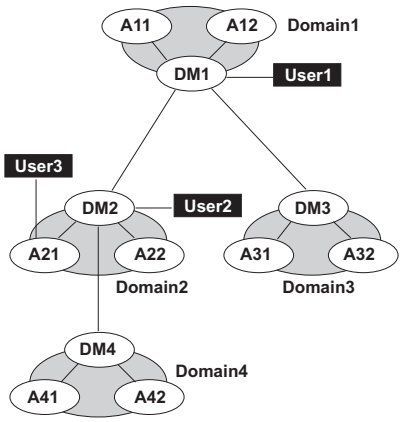


Figura 24. Rede de Exemplo

Tabela 68. Estações de Trabalho Iniciadas

Comando	Iniciado pelo Usuário1	Iniciado pelo Usuário2	Iniciado pelo Usuário3
start @!@	Todas as estações de trabalho são iniciadas	DM2 A21 A22 DM4 A41 A42	A21
start @	DM1 A11 A12	DM2 A21 A22	A21
start DOMAIN3!@	DM3 A31 A32	Não permitido	Não permitido
start DOMAIN4!@	DM4 A41 A42	DM4 A41 A42	Não permitido
start DM2	DM2	DM2	Não permitido
start A42	A42	A42	Não permitido
start A31	A31	Não permitido	Não permitido

startappserver

Inicia o WebSphere Application Server na estação de trabalho.

Sintaxe

startappserver[*domain***!**]*workstation*
[;wait]

Argumentos

domínio

Especifica o nome do domínio da estação de trabalho. Como as estações de trabalho possuem nomes exclusivos, o domínio não é necessário ao iniciar oWebSphere Application Server em uma estação de trabalho específica. Caracteres curinga são permitidos.

Se *domain* for omitido e *workstation* contiver caracteres curinga, o domínio padrão será aquele no qual o **conman** está sendo executado.

estação de trabalho
Especifica o nome da estação de trabalho na qual você deseja iniciar o mecanismo de monitoramento. Caracteres curinga são permitidos. Se nenhum domínio e estação de trabalho estiverem especificados, a ação será na estação de trabalho local.

wait Aguarda até que o WebSphere Application Server tenha sido iniciado antes de solicitar outro comando.

Comentários

A permissão para ações `start` nos objetos `cpu` deve ser ativada no arquivo de segurança para executar este comando.

O WebSphere Application Server também pode ser iniciado com o comando utilitário `StartUp`.

startbrokerapp

Inicia o aplicativo dynamic workload broker.

Sintaxe

startbrokerapp [*domain***!**]*workstation*[*;wait*]

Argumentos

domínio
Especifica o nome do domínio da estação de trabalho. Como as estações de trabalho têm nomes exclusivos, o domínio não é necessário ao iniciar o dynamic workload broker em uma estação de trabalho específica. Caracteres curinga são permitidos.

Se *domain* for omitido e *workstation* contiver caracteres curinga, o domínio padrão será aquele no qual o **conman** está sendo executado.

estação de trabalho
Especifica o nome da estação de trabalho na qual você deseja iniciar o dynamic workload broker. Caracteres curinga são permitidos. Se nenhum domínio e estação de trabalho estiverem especificados, a ação será na estação de trabalho local.

wait Especifica para não aceitar outro comando até que o broker de carga de trabalho dinâmica seja interrompido.

Comentários

A permissão para ações `start` nos objetos `cpu` deve ser ativada no arquivo de segurança para executar este comando.

Também é possível iniciar o dynamic workload broker com wastool `startBrokerApplication`.

starteventprocessor

Inicia o servidor de processamento de eventos no gerenciador de domínio principal, no backup principal ou em uma estação de trabalho instalada como um backup principal que funciona como um agente tolerante a falhas simples.

Sintaxe

{**starteventprocessor** | **startevtp**} [*domain***!**]*workstation*

Argumentos

domínio
Especifica o nome do domínio da estação de trabalho.

estação de trabalho
Especifica o nome da estação de trabalho na qual você deseja iniciar o servidor de processamento de eventos. Caracteres curinga não são permitidos.

Comentários

É possível omitir o nome da estação de trabalho se executar o comando localmente.

A permissão para ações `start` nos objetos `cpu` deve ser ativada no arquivo de segurança para executar este comando.

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Ambiente** > **Monitorar Estações de Trabalho**.
2. Selecione uma tarefa para monitorar estações de trabalho.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. Na tabela que contém a lista de estações de trabalho, selecione uma estação de trabalho e clique em **Mais Ações** > **Iniciar Processador de Evento**.

startmon

Inicia o processo monman que ativa o mecanismo de monitoramento de eventos na estação de trabalho.

Sintaxe

{**startmon** | **startm**} [*domain***!**]*workstation*
[**;noask**]

Argumentos

domínio
Especifica o nome do domínio da estação de trabalho. Como as estações de trabalho possuem nomes exclusivos, o domínio não é necessário ao iniciar o mecanismo de monitoramento em uma estação de trabalho específica. Caracteres curinga são permitidos.

Se *domain* for omitido e *workstation* contiver caracteres curinga, o domínio padrão será aquele no qual o **conman** está sendo executado.

estação de trabalho
Especifica o nome da estação de trabalho na qual você deseja iniciar o mecanismo de monitoramento. Caracteres curinga são permitidos.

noask Especifica para não solicitar confirmação antes da tomada de ação em cada estação de trabalho qualificada.

Comentários

A permissão para ações `start` nos objetos `cpu` deve ser ativada no arquivo de segurança para executar este comando.

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Ambiente** > **Monitorar Estações de Trabalho**.
2. Selecione uma tarefa para monitorar estações de trabalho.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. Na tabela que contém a lista de estações de trabalho, selecione uma estação de trabalho e clique em **Mais Ações** > **Iniciar Monitoramento de Eventos**.

status

Exibe o banner **conman** e o status de produção do Tivoli Workload Scheduler.

Sintaxe

{**status** | **stat**}

Resultados

Depois da palavra **schedule** na segunda linha de saída, o modo de plano de produção (arquivo `Symphony`) é mostrado entre parênteses. As informações **Def** ou **Exp** podem aparecer. **Def** significa que o plano de produção está no modo não expandido, e **Exp** significa que está no modo expandido. O modo do plano de produção é determinado pela definição da opção global ***versão expandida***. Com o Tivoli Workload Scheduler, Versão 8.2, os bancos de dados e os planos são sempre expandidos, mas essas informações aparecem para compatibilidade com versões anteriores.

Exemplos

O exemplo a seguir exibe o status do plano de produção atual.

```
%status
TWS for UNIX/CONMAN 8.4 (1.36.2.22)
Materiais Licenciados Propriedade da IBM
5698-WKB
(C) Copyright IBM Corp 1998, 2007
Direitos Restritos para Usuários do Governo dos Estados Unidos
Uso, duplicação ou divulgação restritos pelo documento
GSA ADP Schedule Contract com a IBM Corp.
Job stream (Exp) 11/26/06 (#34) on site3.
Batchman ATIVO. Limit:19, Fence:0, Audit Level:0
```

stop

Pára os processos de produção do Tivoli Workload Scheduler. Para parar o processo **netman**, utilize o comando **shutdown**. Você deve ter acesso ***stop*** à estação de trabalho.

Sintaxe

stop [*domain***!**]*workstation*
[;wait]
[;noask]

Argumentos

domínio
Especifica o nome do domínio no qual as estações de trabalho pararam. Como as estações de trabalho possuem nomes exclusivos, o domínio não é necessário quando uma determinada estação de trabalho está sendo parada. Caracteres curinga são permitidos.

Este argumento é útil quando se está parando mais de uma estação de trabalho em um domínio. Por exemplo, para parar todos os agentes no domínio**stlouis**, utilize o seguinte comando:

```
stop stlouis!@
```

Se *domain* for omitido e *workstation* contiver caracteres curinga, o domínio padrão será aquele no qual o **conman** está sendo executado.

estação de trabalho
Especifica o nome da estação de trabalho a ser parada. Caracteres curinga são permitidos.

Esse comando não é suportado em estações de trabalho de mecanismo remoto.

wait Especifica para não aceitar outro comando até que todos os processos tenham parado.

noask Especifica para não solicitar confirmação antes da tomada de ação em cada estação de trabalho qualificada.

Comentários

Se o comando **stop** não puder ser aplicado a uma estação de trabalho distante (por exemplo, se o caminho TCP/IP não estiver disponível), o comando será armazenado localmente em um arquivo `pobox` e será enviado para a estação de trabalho quando ela for vinculada.

Supondo-se que um usuário tenha acesso ***stop*** a estações de trabalho sendo paradas, as seguintes regras serão aplicadas:

- Um usuário executando o**conman** no gerenciador de domínio principal pode parar qualquer estação de trabalho na rede.
- Um usuário executando o**conman** em um gerenciador de domínio diferente do principal pode parar qualquer estação de trabalho nesse domínio e em domínios subordinados. O usuário não pode parar estações de trabalho em domínios ponto a ponto.
- Um usuário executando o**conman** em um agente pode parar qualquer estação de trabalho no domínio local.

Quando você emite um comando **stop @** em um gerenciador de domínio, um comando **conman stop** local é executado nas CPUs remotas. O comando inicia a execução nas estações mais baixas na hierarquia da rede e, finalmente, é executado no gerenciador de domínio. No entanto, o arquivo `Symphony` não é atualizado antes das CPUs serem desativadas. Portanto, se você emitir um comando **conman sc@!@** a partir de qualquer CPU, as informações resultantes poderão ser uma figura

atualizada dos estados das CPUs, até mesmo do gerenciador de domínio.

Exemplos

A Figura 25 e a Tabela 69 abaixo mostram a estação de trabalho parada por diferentes comandos **stop** executados por usuários em locais diferentes da rede.

DM*n* são gerenciadores de domínio e **A***nn* são agentes.

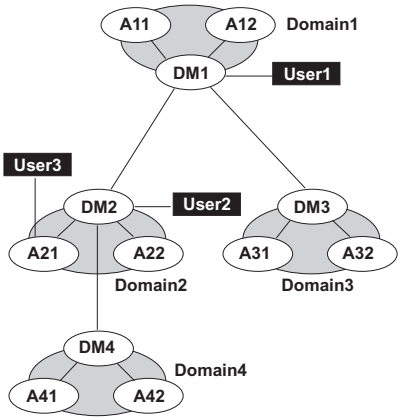


Figura 25. Rede de Exemplo

Tabela 69. Estações de Trabalho Paradas

Comando	Parado pelo: Usuário1	Parado pelo Usuário2	Parado pelo Usuário3
stop @!@	Todas as estações de trabalho são paradas	DM2 A21 A22 DM4 A41 A42	DM2 A21 A22
stop @	DM1 A11 A12	DM2 A21 A22	DM2 A21 A22
stop DOMAIN3!@	DM3 A31 A32	Não permitido	Não permitido
stop DOMAIN4!@	DM4 A41 A42	DM4 A41 A42	Não permitido
stop DM2	DM2	DM2	DM2
stop A42	A42	A42	Não permitido
stop A31	A31	Não permitido	Não permitido

stop ;progressive

Pára os processos de produção do Tivoli Workload Scheduler hierarquicamente quando você tiver definido pelo menos uma estação de trabalho como BEHINDFIREWALL em uma rede do Tivoli Workload Scheduler. Semelhante ao comando stop @!@, mas mais efetivo no aprimoramento do desempenho do plano. O comando não é executado a partir do domínio em que foi inicialmente emitido para cada domínio subordinado, mas em cada nível hierárquico.

Nota: Esse comando não é suportado em estações de trabalho de mecanismo remoto.

Você deve ter acesso ***stop*** à estação de trabalho.

Sintaxe

stop ;progressive

Comentários

Quando você emite o comando em um gerenciador de domínio, todas as estações de trabalho nesse domínio são paradas e, em seguida, o próprio gerenciador de domínio é parado e o comando continua a ser executado em todos os domínios subordinados. A execução do comando continua dessa forma hierárquica, o gerenciador de domínio pára as estações de trabalho no mesmo domínio, ele próprio é parado e, em seguida, a execução continua nos domínios subordinados.

Exemplos

A Figura 26 e a Tabela 70 mostram as estações de trabalho paradas emitindo-se o comando **stop ;progressive** em DM2 e DM4.

DM*n* são gerenciadores de domínio e **A***nn* são agentes.

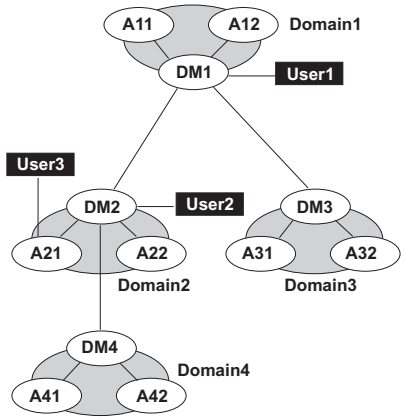


Figura 26. Rede de Exemplo

Tabela 70. Estações de Trabalho Paradas com stop ;progressive

Comando	Parado pelo DM2	Parado pelo DM4
stop ;progressive	A21 A22 DM2	A41 A42 DM4

stopappserver

Para o WebSphere Application Server na estação de trabalho.

Sintaxe

{**stopappserver** | **stopapps**} [*domain***!**]*workstation*
[**;wait**]

Argumentos

domínio
Especifica o nome do domínio da estação de trabalho. Como as estações de trabalho possuem nomes exclusivos, o domínio não é necessário ao parar oWebSphere Application Server em uma estação de trabalho específica. Caracteres curinga são permitidos.

Se *domain* for omitido e *workstation* contiver caracteres curinga, o domínio padrão será aquele no qual o **conman** está sendo executado.

estação de trabalho
Especifica o nome da estação de trabalho na qual você deseja parar o mecanismo de monitoramento. Caracteres curinga são permitidos. Se nenhum domínio e estação de trabalho estiverem especificados, a ação será na estação de trabalho local.

wait Aguarda até que o WebSphere Application Server tenha parado antes de solicitar outro comando.

Comentários

A permissão para as ações `stop` nos objetos `cpu` deve estar ativada no arquivo de segurança para executar este comando.

Nos sistema Windows, não utilize os serviços do Windows para parar o WebSphere Application Server. Se você utilizar esses serviços, o processo *appserverman*, que continua sendo executado, iniciará oWebSphere Application Server novamente. Em vez disso, utilize esse comando ou o comando **stopWas** (sem a opção **-direct**).

Ao executar o comando, o processo *appserverman* primeiro verifica se oWebSphere Application Server pode recuperar as credenciais do usuário (nome de usuário e senha) do arquivo `soap.client.props` localizado no perfil do WebSphere Application Server. Se a verificação for negativa, *appserverman* lê as mesmas do arquivo `useropts` do usuário e executa o script `stopServer.sh (bat)` para passá-las aoWebSphere Application Server.

Para poder executar o comando, você deve, portanto, executar um dos dois procedimentos de customização a seguir para fornecer as credenciais do usuário aoWebSphere Application Server:

- Customize as propriedade de nome de usuário (`com.ibm.SOAP.loginUserid`) e de senha (`com.ibm.SOAP.loginPassword`) no arquivo `soap.client.props` localizado em:

```
WAS_profile_path/properties                    (Version 9.1 and later master and agents)
```

  em que *WAS_profile_path* corresponde ao caminho do perfil do WebSphere Application Server que especificou no momento da instalação. O caminho padrão é `<TWA_home>/WAS/TWSprofile`.

  Você também deve:

  1. Configurar a propriedade `com.ibm.SOAP.securityEnabled` para `true` no mesmo arquivo para ativar a segurança do cliente SOAP
  2. Executar o script **encryptProfileProperties.sh** para criptografar a senha. Consulte o *Guia de Administração do Tivoli Workload Scheduler* para obter mais informações sobre esta ferramenta de servidor de aplicativos.
- Customize a seção `Attributes for conman connections` no arquivo `localopts` especificando os detalhes do conector ou do gerenciador de domínio principal.

Você também deve:

1. Criar (ou customizar se já estiver presente) o arquivo `useropts` manualmente, incluindo os atributos `USERNAME` e `PASSWORD` para o usuário que irá executar `stopappserver`. Certifique-se de que o nome do arquivo `useropts` seja inserido na chave `USEROPTS` na seção `Attributes for conman (CLI) connections`. Consulte o *Guia de Administração do Tivoli Workload Scheduler* para obter mais detalhes.
2. Criptografe a senha no arquivo `useropts` simplesmente executando`conman`.

stopbrokerapp

Para o aplicativo dynamic workload broker.

Sintaxe

{**stopbrokerapp**} [*domain***!**]*workstation*

Argumentos

domínio
Especifica o nome do domínio da estação de trabalho. Como as estações de trabalho têm nomes exclusivos, o domínio não é necessário ao parar o dynamic workload broker em uma estação de trabalho específica. Caracteres curinga são permitidos.

Se *domain* for omitido e *workstation* contiver caracteres curinga, o domínio padrão será aquele no qual o **conman** está sendo executado.

estação de trabalho
Especifica o nome da estação de trabalho na qual você deseja parar o dynamic workload broker. Caracteres curinga são permitidos. Se nenhum domínio e estação de trabalho estiverem especificados, a ação será na estação de trabalho local.

Comentários

A permissão para as ações `stop` nos objetos `cpu` deve estar ativada no arquivo de segurança para executar este comando.

Também é possível parar o dynamic workload broker com o wastool `stopBrokerApplication`.

stopeventprocessor

Pára o servidor de processamento de eventos.

Sintaxe

{**stopeventprocessor** | **stopevtp**} [*domain***!**][*workstation*]

Argumentos

domínio
Especifica o nome do domínio da estação de trabalho.

estação de trabalho
Especifica o nome do gerenciador de domínio principal, do backup principal ou da estação de trabalho instalada como um backup principal

que funciona como um agente tolerante a falhas simples no qual você deseja parar o servidor de processamento de eventos. Caracteres curinga não são permitidos.

É possível omitir o nome da estação de trabalho se executar o comando localmente.

Comentários

Este comando não pode ser emitido de uma maneira assíncrona.

Se você emitir o comando de uma estação de trabalho diferente daquela onde o processador de eventos está configurado, o comando utilizará o cliente da linha de comandos e, dessa forma, as credenciais de usuários para o cliente de linha de comandos devem estar configuradas corretamente.

A permissão para as ações `stop` nos objetos `cpu` deve estar ativada no arquivo de segurança para executar este comando.

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Ambiente** > **Monitorar Estações de Trabalho**.
2. Selecione uma tarefa para monitorar estações de trabalho.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. Na tabela que contém a lista de estações de trabalho, selecione uma estação de trabalho e clique em **Mais Ações** > **Parar Processador de Evento**.

stopmon

Pára o mecanismo de monitoramento de eventos na estação de trabalho.

Sintaxe

{**stopmon** | **stopm**} [*domain***!**]*workstation*
[**;wait**]
[**;noask**]

Argumentos

domínio
Especifica o nome do domínio da estação de trabalho. Como as estações de trabalho possuem nomes exclusivos, o domínio não é necessário ao parar o mecanismo de monitoramento em uma estação de trabalho específica. Caracteres curinga são permitidos.

Se *domain* for omitido e *workstation* contiver caracteres curinga, o domínio padrão será aquele no qual o **conman** está sendo executado.

estação de trabalho
Especifica o nome da estação de trabalho na qual você deseja parar o mecanismo de monitoramento. Caracteres curinga são permitidos.

wait Especifica para não aceitar outro comando até que o mecanismo de monitoramento seja parado.

noask Especifica para não solicitar confirmação antes da tomada de ação em cada estação de trabalho qualificada.

Comentários

O mecanismo de monitoramento é reiniciado automaticamente quando o próximo plano de produção é ativado (no Windows também quando oTivoli Workload Scheduler é reiniciado) a menos que você desative a opção local `autostart monman`.

O comando é assíncrono, a menos que você especifique a palavra-chave `wait`.

A permissão para as ações `stop` nos objetos `cpu` deve estar ativada no arquivo de segurança para executar este comando.

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Ambiente** > **Monitorar Estações de Trabalho**.
2. Selecione uma tarefa para monitorar estações de trabalho.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. Na tabela que contém a lista de estações de trabalho, selecione uma estação de trabalho e clique em **Mais Ações** > **Parar Monitoramento de Eventos**.

submit docommand

Envia um comando a ser ativado como uma tarefa.

Para executar este comando, no arquivo de segurança você deve ter acesso de ***envio*** para a tarefa com o nome especificado em sua definição de banco de dados e, se você utilizar a palavra-chave ***alias***, também com o nome especificado com esta palavra-chave. Além disso, se você utilizar a palavra-chave ***recoveryjob***, deverá ter acesso de ***envio*** para a tarefa especificada com essa palavra-chave.

Para incluir dependências `needs` e `prompt`, você deve ter acesso ***use*** aos recursos e prompts globais.

Se você enviar a tarefa de uma estação de trabalho diferente do gerenciador de domínio principal, deverá estar conectando como um usuário que:

- possui credenciais adequadas definidas no arquivo arquivo useropts para conectar-se aogerenciador de domínio principal por meio do WebSphere Application Server
- seja autorizado a executar comandos de envio no arquivo de segurança armazenado no gerenciador de domínio principal

Sintaxe

{**submit docommand** | **sbd**} [*workstation*#]"*cmd*"
[**;alias**[**=***name*]]
[**;into=**[*workstation*#]
{*jobstream_id***;schedid** |*jobstreamname* ([*hhmm*[*date*]])}]
[**;***joboption*[**;**...]]

Argumentos

estação de trabalho
Especifica o nome da estação de trabalho na qual o job será lançado. Caracteres curinga são permitidos e, nesse caso, o job é lançado em todas as estações de trabalho qualificadas. O padrão é a estação de trabalho na qual o **conman** está sendo executado. Você não pode especificar um domínio ou classe de estação de trabalho.

Nota: Em razão de uma limitação na maneira com que o Windows gerencia o sinal de igual (=) no ambiente shell, é necessário mascarar o sinal de igual (=) desta forma '\='\ ao enviar comandos do Windows utilizando**submit docommand**. Por exemplo, para configurar a variável local **var1** como *hello*, é necessário emitir o seguinte comando:

```
%sbd "set var1\"=\"hello"
```

cmd Especifica um comando de sistema válido de até 255 caracteres. Todo o comando deve ser colocado entre aspas ("). O comando será tratado como um job e todas as regras de job serão aplicadas.

alias=*name*
Especifica um nome exclusivo a ser atribuído ao job. Se você inserir a palavra-chave **alias** sem especificar um nome, um nome será construído utilizando até os seis primeiros caracteres alfanuméricos (em letras maiúsculas) do comando, dependendo do número de caracteres no comando, seguido de um número aleatório de dez dígitos. Se houver espaços em branco no comando, o nome será construído utilizando até os seis primeiros caracteres alfanuméricos antes do espaço em branco. Por exemplo, se o comando for **"rm apfile"**, o nome gerado será semelhante a **RM0123456789**. Se o comando for mais longo do que seis caracteres alfanuméricos como, **"wlsinst"**, o nome gerado será **wlsins0396578515**.

Se você não incluir um **alias** na primeira vez que submeter o comando, um nome de job será construído utilizando até 255 caracteres do nome do comando. Se você submeter um comando uma segunda vez a partir da mesma estação de trabalho, a palavra-chave **alias** será obrigatória e deverá ser exclusiva para cada submissão do comando.

into=*jobstream_instance*
Identifica a instância do fluxo de tarefas na qual a tarefa será colocada para ativação. Selecione a instância do fluxo de tarefas da seguinte forma:

[*workstation***#**]*jobstreamname*([*hhmm*[*date*]])

ou

[*workstation***#**]*jobstream_id* **;schedid**

Se **into** não for utilizado, o job será adicionado a um fluxo de job chamado**JOBS**.

joboption
Especifique qualquer um dos seguintes:

at=*hhmm* [**timezone**|**tz** *tzname*] [**+***n* **days** | *mm/dd[/yy]*] | [**absolute** | **abs**]

confirmed

critical

deadline=*time* [**timezone**|**tz** *tzname*][**+***n***day**[**s** | *mm/dd[/yy]*]

every=*rate*

follows=[*netagent*::][*workstation*#]{*jobstreamname*[*hhmm* [*mm/dd[/yy]*]][.*job* | @] | *jobstream_id.job*;**schedid**} | *job*[**;nocheck**][**;wait**=*time*][**,**...]

Nota: O argumento **;nocheck** não é suportado nas dependências da interligação de redes.

interactive

Nota: Esta palavra-chave pode ser utilizada somente nos ambientes Windows.

maxdur=*time*[**onmaxdur** *action*]

mindur=*time*[**onmindur** *action*]

logon=*user*.

needs=[*num*] [*workstation*#]*resource*[**,**...]

opens=[*workstation*#]"*filename*"[**(***qualifier***)**][**,**...]

priority=[*pri* | **hi** | **go**]

prompt="[**:** | **!**]*text***"** | *promptname*[**,**...]

rccondsucc"*Success Condition*"

recovery=stop | continue | rerun

recoveryjob=[*workstation*#]*jobname*

O nome de uma tarefa de recuperação diferente de uma (se presente) especificada na definição de tarefa no banco de dados.

after [*workstation*#]*jobname*

abendprompt *"text"*

until *time* [**timezone | tz** *tzname*][+*n* **day**[**s**] | [**absolute** | **abs**]] [**;onuntil** *action*]

O valor padrão para ***joboption*** é o usuário na estação de trabalho a partir da qual o comando está sendo executado.

Utilizando Parâmetros Locais

É possível utilizar parâmetros locais como valores com as seguintes palavras-chave:

- `cmd`
- `opens`
- `logon`
- `prompt`
- `abendprompt`

Os parâmetros locais são definidos e gerenciados com o comando de utilitário parms em um banco de dados local na estação de trabalho na qual a tarefa é executada. Os parâmetros são resolvidos na estação de trabalho enquanto o comando **`submit`** está em execução.

Comentários

Tarefas enviadas na produção a partir da linha de comandos **conman** não são incluídas no plano de pré-produção e, portanto, não podem ser levadas em consideração durante a identificação de predecessores de dependências de seqüências externas.

Se você não especificar uma *estação de trabalho* com `follows`, `needs` , `opens` ou `into`, o padrão será a estação de trabalho da tarefa.

O planejador classifica dependências follows como *internas* quando são especificadas apenas por seu nome da tarefa dentro do fluxo de tarefas. Ele classifica-as como *externas* quando são especificadas no formato *jobStreamName.workstationName.jobName*.

Ao enviar o objeto em um fluxo de tarefas e incluir uma dependência de sequência que compartilhe o mesmo nome de fluxo de tarefas (por exemplo, você envia o objeto no fluxo de tarefas `schedA` e define uma dependência de sequência em `schedA.job2`), a dependência é tratada como uma dependência de sequência *externa*. Desde a Versão 8.3, ao contrário das versões anteriores, porque o planejador usa os critérios de correspondência `mesmo dia` para resolver dependências externas, dependências originadas dessa maneira nunca são incluídas da primeira vez que o objeto é enviado.

Exemplos

Para submeter um comando **rm** ao fluxo de job `JOBS` com uma dependência `follows`, execute o seguinte comando:

```
submit docommand="rm apfile";follows sked3
```

Para submeter um comando **sort** com o alias **sortit** e colocar o job no fluxo de job `reports` com uma hora **at** de 17h30, execute o seguinte comando:

```
sbd "sort < file1 > file2";alias=sortit;into=reports;at=1730
```

Para submeter comandos **chmod** em todas as estações de trabalho com nomes começando com `site`, execute o seguinte comando:

```
sbd site@#"chmod 444 file2";alias
```

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

oGuia do Usuário do Console do Dynamic Workload, na Enviando Tarefas Ad Hoc.

submit file

Envia um arquivo a ser ativado como uma tarefa.

Para executar este comando, no arquivo de segurança você deve ter acesso de ***envio*** para a tarefa com o nome especificado em sua definição de banco de dados e, se você utilizar a palavra-chave ***alias***, também com o nome especificado com esta palavra-chave. Além disso, se você utilizar a palavra-chave ***recoveryjob***, deverá ter acesso de ***envio*** para a tarefa especificada com essa palavra-chave.

Para incluir dependências `needs` e `prompt`, você deve ter acesso ***use*** aos recursos e prompts globais.

Se você enviar a tarefa de uma estação de trabalho diferente do gerenciador de domínio principal, deverá estar conectando como um usuário que:

- Possui credenciais adequadas definidas no arquivo useropts para conectar-se ao gerenciador de domínio principal por meio do WebSphere Application Server.
- Seja autorizado a executar comandos de envio no arquivo de segurança armazenado no gerenciador de domínio principal

Sintaxe

{**submit file** | **sbf**} "*filename*"
[**;alias**[=*name*]]
[**;into=**[*workstation***#**]{*jobstream_id*
;schedid |*jobstreamname*([*hhmm*[*date*]])}]
[**;***joboption*[**;**...]]
[**;noask**]

Argumentos

nome do arquivo
Especifica o nome do arquivo, até 255 caracteres. Caracteres curinga são permitidos. O nome deverá ser colocado entre aspas (") se contiver caracteres diferentes de caracteres alfanuméricos, traços (-), barras (/) e sublinhados (_). Consulte os exemplos.

alias=*name*
Especifica um nome exclusivo a ser atribuído ao job. Se você inserir a palavra-chave **alias** sem especificar um nome, um nome será construído utilizando até os seis primeiros caracteres alfanuméricos (em letras maiúsculas) do nome do arquivo, dependendo do número de caracteres no nome do arquivo, seguido de um número aleatório de dez dígitos. Por exemplo, se o nome do arquivo for `jclttx5`, o nome gerado será semelhante a `JCLTTX0123456789`.

Se você não incluir **alias**, um nome de arquivo é construído usando até 255 caracteres alfanuméricos do nome base do arquivo em maiúscula.

Em qualquer um dos casos acima, se o nome do arquivo não iniciar com uma letra, você será solicitado a utilizar **alias=** *name*.

Se você submeter um arquivo uma segunda vez a partir da mesma estação de trabalho, a palavra-chave **alias** será obrigatória e deverá ser exclusiva para cada submissão do arquivo.

into=*jobstream_instance*
Identifica a instância do fluxo de tarefas na qual a tarefa será colocada para ativação. Selecione a instância do fluxo de tarefas da seguinte forma:

[*workstation***#**]*jobstreamname*([*hhmm*[*date*]])

ou

[*workstation***#**]*jobstream_id* **;schedid**

Se **into** não for utilizado, o job será adicionado a um fluxo de job chamado**JOBS**.

joboption
Especifique um dos seguintes:

at=*hhmm* [**timezone**|**tz** *tzname*] [**+***n* **days** | *mm/dd[/yy]*] | [**absolute** | **abs**]

confirmed

critical

deadline=*time*[**timezone** | **tz** *tzname*][**+***n* **days** | *mm/dd[/yy]*]

every=*rate*

follows=[*netagent***::**][*workstation***#**]{*jobstreamname*(*hhmm* [*mm/dd[/yy]*])[*.job* | **@**] | *jobstream_id.job***;schedid**} | *job*[**;nocheck**][**;wait=***time*][**,**...]

Nota: O argumento **;nocheck** não é suportado nas dependências da interligação de redes.

interactive

Nota: Esta palavra-chave pode ser utilizada somente nos ambientes Windows.

logon=*user*

maxdur=*time*[**onmaxdur** *action*]

mindur=*time*[**onmindur** *action*]

needs=[*num*] [*workstation***#**]*resource*[**,**...]

opens=[*workstation***#**]"*filename*"[**(***qualifier***)**][**,**...]

priority=[*pri* | **hi** | **go**]

prompt="[: | **!**]*text***"** | *promptname*[**,**...]

rccondsucc"*Success Condition*"

recovery=stop | **continue** | **rerun**

recoveryjob=[*workstation***#**]*jobname*

O nome de uma tarefa de recuperação diferente de uma (se presente) especificada na definição de tarefa no banco de dados.

after [*workstation***#**]*jobname*

abendprompt *"text"*

until *time* [**timezone**|**tz** *tzname*][**+***n* **day**[**s**] | [**absolute** | **abs**]] [**;onuntil** *action*]

noask Especifica para não solicitar confirmação antes da tomada de ação em cada arquivo qualificado.

Utilizando Parâmetros Locais

É possível utilizar parâmetros locais como valores com as seguintes palavras-chave:

- `opens`
- `logon`
- `prompt`
- `abendprompt`

Os parâmetros locais são definidos e gerenciados com o comando de utilitário parms em um banco de dados local na estação de trabalho na qual a tarefa é

executada. Os parâmetros são resolvidos na estação de trabalho enquanto o comando **submit** está em execução.

Comentários

Tarefas enviadas na produção a partir da linha de comandos **conman** não são incluídas no plano de pré-produção e, portanto, não podem ser levadas em consideração durante a identificação de predecessores de dependências de seqüências externas.

Se você não especificar uma estação de trabalho com `follows`, `needs` , `opens` ou `into`, o padrão será a estação de trabalho em que **conman** está em execução.

O planejador classifica dependências follows como *internas* quando são especificadas apenas por seu nome da tarefa dentro do fluxo de tarefas. Ele classifica-as como *externas* quando são especificadas no formato *jobStreamName.workstationName.jobName*.

Ao enviar o objeto em um fluxo de tarefas e incluir uma dependência de sequência que compartilhe o mesmo nome de fluxo de tarefas (por exemplo, você envia o objeto no fluxo de tarefas `schedA` e define uma dependência de sequência em `schedA.job2`), a dependência é tratada como uma dependência de sequência *externa*. Desde a Versão 8.3, ao contrário das versões anteriores, porque o planejador usa os critérios de correspondência `mesmo dia` para resolver dependências externas, dependências originadas dessa maneira nunca são incluídas da primeira vez que o objeto é enviado.

Exemplos

Para submeter um arquivo ao fluxo de job `jobs` (o nome do job é `myjcl`), execute o seguinte comando:

```
submit file=d:\jobs\lib\daily\myjcl
```

em que a seqüência `;into` foi omitida.

Para submeter um arquivo com um nome de job de `misjob4` ao fluxo de job `missked`, execute o seguinte comando:

```
sbf /usr/lib/mis/jcl4;alias=misjob4;into=missked ;needs=2 slots
```

O job precisa de duas unidades do recurso`slots`.

Para submeter todos os arquivos com nomes começando com `back` ao fluxo de job `bkup`, execute o seguinte comando:

```
sbf "/usr/lib/backup/back@";into=bkup
```

Para enviar o arquivo `tws_env.cmd`, cujo caminho contém um espaço vazio, em uma estação de trabalho do Windows, execute:

- No modo interativo:

  ```
  sbf "\"C:\Program Files\IBM\TWS\lucaMDM\tws_env.cmd\"";alias=MYJOB
  ```

  Estando no Windows, as aspas duplas (") devem ser escapadas pela sequência de caracteres `"\`.
- No modo de linha de comandos:

  ```
  conman sbf "\"\\\"C:\Program Files\IBM\TWS\lucaMDM\tws_env.cmd\\\"\"";alias=MYJOB
  ```

  Estando no Windows e executando o comando externamente a partir do ambiente do conman, a seqüência da função escape torna-se mais longa.

em que "\" é o caractere de escape para o espaço em branco no caminho de arquivo.

submit job

Envia uma tarefa para ser ativada.

Para executar esse comando, no arquivo de segurança você deve ter acesso de ***envio*** (***submitdb***) para a tarefa com o nome especificado em sua definição de banco de dados e, se usar a palavra-chave ***alias***, também com o nome especificado com esta palavra-chave. Além disso, se você utilizar a palavra-chave ***recoveryjob***, deverá ter acesso de ***envio*** para a tarefa especificada com essa palavra-chave.

Observe que se você tiver somente direitos de segurança ***submitdb***, você está limitado a enviar tarefas definidas no banco de dados. Não é possível enviar tarefas ad hoc.

Para incluir dependências `needs` e `prompt`, você deve ter acesso ***use*** aos recursos e prompts globais.

Se você enviar a tarefa de uma estação de trabalho diferente do gerenciador de domínio principal, deverá estar conectando como um usuário que:

- Possui credenciais adequadas definidas no arquivo useropts para conectar-se aogerenciador de domínio principal por meio do WebSphere Application Server
- Esteja autorizado a executar comandos **submit** no arquivo de segurança armazenado no gerenciador de domínio principal

Se uma tarefa sombra for enviada, consulte Capítulo 19, "Definindo e Gerenciando Dependências Cruzadas", na página 625 para obter detalhes adicionais.

Sintaxe

{**submit job** | **sbj**} [*workstation*#]*jobname*
[**;alias**[**=***name*]]
[**;into=**[*workstation***#**]{*jobstream_id*
;schedid |*jobstreamname*([*hhmm*[*date*]])}]
[**;***joboption*[**;**...]]
[**;vartable=***tablename*]
[**;noask**]

Argumentos

estação de trabalho
Especifica o nome da estação de trabalho na qual o job será lançado. Caracteres curinga são permitidos e, nesse caso, o job é lançado em todas as estações de trabalho qualificadas. O padrão é a estação de trabalho na qual o `conman` está sendo executado. Você não pode especificar um domínio ou classe de estação de trabalho.

jobname
Especifica o nome do job. Caracteres curinga são permitidos e, nesse caso, todos os jobs qualificados são submetidos. Se o job já estiver no plano de produção e estiver sendo submetido ao mesmo fluxo de job, você deverá utilizar o argumento **alias** para atribuir um nome exclusivo.

alias=*name*
Especifica um nome exclusivo a ser atribuído ao job no lugar de *jobname*.

Se você inserir a palavra-chave **alias** sem especificar um nome, um nome será construído usando os cinco primeiros caracteres alfanuméricos de *jobname* seguidos por um número aleatório de dez dígitos. O nome é sempre organizado em ordem crescente. Por exemplo, se *jobname* for `jcrttx5`, o nome gerado será semelhante a `JCRTT1234567890`.

into=*jobstream_instance*
Identifica a instância do fluxo de tarefas na qual a tarefa será colocada para ativação. Selecione a instância do fluxo de tarefas da seguinte forma:

[*workstation***#**]*jobstreamname*([*hhmm*[*date*]])

ou

[*workstation***#**]*jobstream_id* **;schedid**Se **into** não for utilizado, o job será adicionado a um fluxo de job chamado**JOBS**.

joboption
Especifique um dos seguintes:

at=*hhmm* [**timezone**|**tz** *tzname*] [+*n* **days** | *mm/dd[/yy]*] | [**absolute** | **abs**]

confirmed

critical

deadline=*time*[**timezone** | **tz** *tzname*][+*n* **days** | *mm/dd[/yy]*]

every=*rate*

follows=[*netagent***::**][*workstation***#**]{*jobstreamname*(*hhmm* [*mm/dd[/yy]*])[*.job* | **@**] | *jobstream_id.job***;schedid**} | *job*[**;nocheck**][**;wait**=*time*][**,**...]

Nota: O argumento **;nocheck** não é suportado nas dependências da interligação de redes.

maxdur=*time*[**onmaxdur** *action*]

mindur=*time*[**onmindur** *action*]

needs=[*num*] [*workstation***#**]*resource*[**,**...]

opens=[*workstation***#**]"*filename*"[**(***qualifier***)**][**,**...]

priority=[*pri* | **hi** | **go**]

prompt="[**:** | **!**]*text***"** | *promptname*[**,**...]

rccondsucc"*Success Condition*"

recovery=stop | **continue** | **rerun**

recoveryjob= [*workstation***#**]*jobname*

O nome de uma tarefa de recuperação diferente de uma (se presente) especificada na definição de tarefa no banco de dados.

after [*workstation***#**]*jobname*

abendprompt "*text*"

until *time* [**timezone**|**tz** *tzname*][+*n* **day**[**s**] | [**absolute** | **abs**]] [**;onuntil** *action*]

vartable=*tablename*
Especifica o nome da tabela de variáveis, se for diferente do padrão, em que as variáveis que deseja usar são definidas.

Lembre-se:

- Com esse comando, você pode utilizar a substituição de variáveis para as seguintes palavras-chave:
  - `opens`
  - `prompt`
  - `abendprompt`
- Coloque a variável entre acentos circunflexos (^) e a cadeia completa entre aspas. Se a variável contiver uma parte de um caminho, certifique-se de que os caracteres acento circunflexo não sejam imediatamente precedidos por uma barra invertida (\), pois nesse caso, a seqüência \^ poderia ser interpretada erroneamente como uma seqüência de escape e resolvida pelo analisador como caractere acento circunflexo. Se necessário, mova a barra invertida para a definição da variável entre acentos circunflexos.
- As variáveis especificadas na definição de tarefa com o formato ${*variablename*} não são resolvidas.

Se você submeter uma tarefa contendo variáveis definidas em uma tabela de variáveis que não seja a tabela de variáveis padrão e não especificar a tabela de variáveis no ciclo de execução, no fluxo de tarefas ou na estação de trabalho, as variáveis não serão resolvidas. Consulte o Capítulo 6, "Customizando a Carga de Trabalho Utilizando Tabelas de Variáveis", na página 113.

noask Especifica para não solicitar confirmação antes da tomada de ação em cada job qualificado.

Comentários

Tarefas enviadas na produção a partir da linha de comandos **conman** não são incluídas no plano de pré-produção e, portanto, não podem ser levadas em consideração durante a identificação de predecessores de dependências de seqüências externas.

Se você não especificar uma estação de trabalho com `follows`, `needs` , `opens` ou `into`, o padrão será a estação de trabalho da tarefa.

at especifica em que horário a tarefa pode ser enviada. Se a palavra-chave **at** for utilizada, a tarefa não poderá iniciar antes do horário configurado nessa palavra-chave. Observe que se o gerenciador de domínio principal da rede executar com as opções `enLegacyStartOfDayEvaluation` e `enTimeZone` configuradas para `yes` para converter o horário `startOfDay` configurado no gerenciador de domínio principal para o fuso horário local configurado em cada estação de trabalho na rede, você deverá incluir a palavra-chave **absolute** para que funcione.

O planejador classifica dependências follows como *internas* quando são especificadas apenas por seu nome da tarefa dentro do fluxo de tarefas. Ele classifica-as como *externas* quando são especificadas no formato *jobStreamName.workstationName.jobName*.

Ao enviar o objeto em um fluxo de tarefas e incluir uma dependência de sequência que compartilhe o mesmo nome de fluxo de tarefas (por exemplo, você envia o objeto no fluxo de tarefas `schedA` e define uma dependência de sequência em `schedA.job2`), a dependência é tratada como uma dependência de sequência *externa*. Desde a Versão 8.3, ao contrário das versões anteriores, porque o planejador usa os critérios de correspondência `mesmo dia` para resolver

dependências externas, dependências originadas dessa maneira nunca são incluídas da primeira vez que o objeto é enviado.

Exemplos

Para submeter os jobs `test` ao fluxo de job `JOBS`, execute o seguinte comando:

```
sbj test
```

Para submeter um job com um alias de `rptx4` e colocar o job no fluxo de job `reports` com uma hora **at** de 5h30, execute o seguinte comando:

```
sbj rjob4;alias=rptx4;into=reports;at=1730
```

Para submeter o job `txjob3` a todas as estações de trabalho cujos nomes comecem com `site`, execute o seguinte comando:

```
sbj site@#txjob3;alias
```

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

oGuia do Usuário do Console do Dynamic Workload, seção Enviando Tarefas Predefinidas.

submit sched

Envia um fluxo de tarefa para processamento.

Para executar este comando, no arquivo de segurança, você deve ter acesso de ***envio*** para o fluxo de tarefa com o nome especificado em sua definição de banco de dados e, se você utilizar a palavra-chave ***alias***, também com o nome especificado com esta palavra-chave. Para incluir dependências `needs` e `prompt`, você deve ter acesso ***use*** aos recursos e prompts globais.

O comando `submit schedule` utiliza as credenciais configuradas no arquivo `useropts` que pertencem ao*TWS_user* que instalou essa estação de trabalho.

Se você enviar o fluxo de tarefas de uma estação de trabalho diferente de gerenciador de domínio principal, deverá estar conectando como um usuário que:

- possui credenciais adequadas definidas no arquivo arquivo useropts para conectar-se aogerenciador de domínio principal por meio do WebSphere Application Server
- seja autorizado a executar comandos de envio no arquivo de segurança armazenado no gerenciador de domínio principal

Sintaxe

{**submit sched** | **sbs**} [*workstation*#]*jstreamname*
[**;alias**[**=***name*]]
[*;jstreamoption*[**;**...]]
[**;vartable=***tablename*]
[**;noask**]

Argumentos

estação de trabalho
Especifica o nome da estação de trabalho na qual fluxo de job será lançado.

Caracteres curinga são permitidos e, nesse caso, o fluxo de job é lançado em todas as estações de trabalho qualificadas. O padrão é a estação de trabalho na qual o **conman** está sendo executado. Você não pode especificar um domínio ou classe de estação de trabalho.

jstreamname
Especifica o nome do fluxo de tarefas. Caracteres curinga são permitidos e, nesse caso, todos os fluxos de job qualificados são submetidos. Se o fluxo de job já estiver no plano de produção, você deverá utilizar o argumento **alias** para atribuir um nome exclusivo.

alias=*name*
Especifica um nome exclusivo a ser atribuído ao fluxo de job no lugar de *jstreamname*. Se configurado, esse valor também corresponde ao*jobstream_id*. Se você inserir a palavra-chave **alias** sem especificar um nome, será construído um nome usando os cinco primeiros caracteres alfanuméricos de *jstreamname* seguidos por um número aleatório de dez dígitos. O nome é sempre organizado em ordem crescente. Por exemplo, se *jstreamname* for `sttrom`, o nome gerado será semelhante a `STTRO1234567890`.

A autorização para enviar o planejamento é marcada no arquivo de Segurança usando o nome original, não o nome alternativo.

jstreamoption
Insira qualquer um dos seguintes (consulte "Detalhes da Palavra-chave de Definição do Fluxo de Tarefas" na página 220 para localizar quais opções são mutuamente exclusivas):

[at=*hhmm* **[timezone|tz** *tzname***] [+***n* **days** | *date***] [absolute | abs]] | [schedtime=[***hhmm* **[***date***] | [+***n* **days]]**
em que:

at especifica o horário em que o fluxo de tarefas pode ser ativado. Se a palavra-chave palavra-chave at for usada, o fluxo de tarefas não poderá iniciar antes do horário configurado com esta palavra-chave (consulte o tópico sobre as palavras-chave de definição de fluxo de tarefas no capítulo "Definindo Objetos no Banco de Dados" no "*Guia do Usuário e Referência do IBM Tivoli Workload Scheduler*" para obter informações adicionais sobre a palavra-chave "at"). Observe que se o gerenciador de domínio principal da rede executar com as opções `enLegacyStartOfDayEvaluation` e `enTimeZone` configuradas para `yes` para converter o horário `startOfDay` configurado no gerenciador de domínio principal para o fuso horário local configurado em cada estação de trabalho na rede, você deverá incluir a palavra-chave **absolute** para que funcione.

schedtime representa o dia e o horário em que o fluxo de tarefas é posicionado no plano. Se, neste horário, o fluxo de tarefas estiver livre de dependências e não tiver restrições de tempo**at** definidas, ele será ativado. Um valor designado para **schedtime** não representa uma dependência para o fluxo de tarefas. Seu valor é, então, exibido nas colunas *SchedTime* na saída dos comandos show. Se uma restrição **at** for definida, o valor designado para **schedtime** será sobrescrito pelo valor **at**. Quando o fluxo de tarefas realmente inicia, o valor designado para **schedtime** é sobrescrito pelo horário de início real do fluxo de tarefas.

O formato utilizado para *date* depende do valor designado para a variável ***date format*** especificada no arquivo `localopts`.

Se nenhum fuso horário adicional for especificado, o fuso horário configurado na estação de trabalho que executa o comando será assumido.

carryforward

deadline=*time***[timezone | tz** *tzname***][+***n* **days | ***date***]**
Se nenhum fuso horário adicional for especificado, o fuso horário configurado na estação de trabalho que executa o comando será assumido.

follows=[*netagent***::][***workstation***#]{***jobstreamname***[***hhmm* **[***mm/dd[/yy]***]][.***job* **| @] | *jobstream_id*.*job*;schedid}| *job*[;nocheck][;wait=***time***][,...]**
Os critérios de correspondência utilizados ao enviar fluxos de tarefa na produção são diferentes da maneira como as dependências **follows** são resolvidas no plano de pré-produção. Quando um fluxo de tarefas, por exemplo, JS_A, contendo uma dependência **follows** de uma tarefa ou um fluxo de tarefa, por exemplo, JS_B, é enviado a partir de um programa de linha de comandos **conman**, a instância predecessora de JS_B é definida seguindo estes critérios:

1. A instância mais próxima de JS_B antes de JS_A.
2. Se não existir nenhuma instância anterior a JS_B, a instância predecessora será a instância mais próxima de JS_B depois de JS_A.
3. Caso contrário, um erro será exibido e o comando falhará se a palavra-chave **;nocheck** não for utilizada.

A instância do fluxo de tarefas do predecessor é procurada entre as instâncias incluídas no plano de produção quando**JnextPlan** foi executado e as instâncias foram enviadas em produção com o comando **sbs**, incluindo as enviadas com um alias.

Atenção: O argumento **;nocheck** não é suportado nas dependências da interligação de redes.

limit=*joblimit*

needs=[*num***] [***workstation***#]***resource***[,...]**

opens=[*workstation***#]"***filename***"[(***qualifier***)][,...]**

priority=[*pri* **| hi | go]**

prompt="[: | !]*text***" | *promptname***[,...]**

until *time* **[timezone|tz** *tzname***][+***n* **day[s] | [absolute | abs]] [;onuntil** *action***]**

Se nenhum fuso horário adicional for especificado, o fuso horário configurado na estação de trabalho que executa o comando será assumido.

vartable=*tablename*
Especifica o nome da tabela de variáveis, se for diferente do padrão, em que as variáveis que deseja usar são definidas.

Lembre-se:

- Com esse comando, você pode utilizar a substituição de variáveis para as seguintes palavras-chave:
  - opens

– prompt

- Coloque a variável entre acentos circunflexos (^) e a cadeia completa entre aspas. Se a variável contiver uma parte de um caminho, certifique-se de que os caracteres acento circunflexo não sejam imediatamente precedidos por uma barra invertida (\), pois nesse caso, a seqüência \^ poderia ser interpretada erroneamente como uma seqüência de escape e resolvida pelo analisador como caractere acento circunflexo. Se necessário, mova a barra invertida para a definição da variável entre acentos circunflexos.

Se você submeter um fluxo de tarefas com tarefas contendo variáveis definidas em uma tabela de variáveis que não seja a tabela de variáveis padrão e não especificar a tabela de variáveis no ciclo de execução, no fluxo de tarefas ou na estação de trabalho, as variáveis não serão resolvidas. Consulte o Capítulo 6, "Customizando a Carga de Trabalho Utilizando Tabelas de Variáveis", na página 113.

noask Especifica para não solicitar confirmação antes da tomada de ação em cada fluxo de job qualificado.

Comentários

Os fluxos de tarefas enviados na produção a partir da linha de comandos **conman** não são incluídos no plano de pré-produção e, portanto, não podem ser levados em consideração durante a identificação de predecessores de dependências follows externas.

Se você não especificar uma estação de trabalho com `follows`, `needs` ou `opens`, o padrão será a estação de trabalho do fluxo de job.

O planejador classifica dependências follows como *internas* quando são especificadas apenas por seu nome da tarefa dentro do fluxo de tarefas. Ele classifica-as como *externas* quando são especificadas no formato *jobStreamName.workstationName.jobName*.

Ao enviar um fluxo de tarefas que inclui uma tarefa com uma dependência de sequência que compartilha o mesmo nome de fluxo de tarefas (por exemplo, o fluxo de tarefas `schedA` inclui uma tarefa denominada `job6` que tem uma dependência de sequência em `schedA.job2`), a dependência é incluída como uma dependência de sequência *externa*. Desde a Versão 8.3, ao contrário das versões anteriores, porque o planejador usa os critérios de correspondência `mesmo dia` para resolver dependências externas, dependências originadas dessa maneira nunca são incluídas da primeira vez que o objeto é enviado.

Exemplos

Para submeter o fluxo de job `adhoc` à estação de trabalho`site1` e sinalizá-la como um fluxo de job **carryforward**, execute o seguinte comando:

```
submit sched=site1#adhoc;carryforward
```

Para submeter o fluxo de job `fox4` com um limite de job de **2**, uma prioridade de **23** e uma hora **until** de meia-noite, execute o seguinte comando:

```
sbs fox4;limit=2;pri=23;until=0000
```

Para submeter o fluxo de job `sched3` a todas as estações de trabalho com nomes iniciados com `site`, execute o seguinte comando:

```
sbs site@#sched3
```

Consulte também

A partir do Dynamic Workload Console, é possível executar a mesma tarefa, conforme descrito em:

oGuia do Usuário do Console do Dynamic Workload, seção Enviando Fluxos de Tarefas Predefinidos.

switcheventprocessor

Comuta o servidor de processamento de eventos do gerenciador de domínio principal para o backup principal ou vice-versa.

Observe que é possível executar o servidor de processamento de eventos também em uma estação de trabalho instalada como um backup principal que é executado como um agente tolerante a falhas simples.

Sintaxe

{**switcheventprocessor** | **switchevtp**} *workstation*

Argumentos

estação de trabalho
Especifica o nome do gerenciador de domínio principal ou do backup principal no qual você deseja comutar o servidor de processamento de eventos. Caracteres curinga não são permitidos.

Comentários

Se você emitir o comando de uma estação de trabalho diferente daquela onde o processador de eventos está configurado, o comando utilizará o cliente da linha de comandos e, dessa forma, as credenciais de usuários para o cliente de linha de comandos devem estar configuradas corretamente.

No caso de principais de backup, a estação de trabalho deve ter o atributo`full-status` configurado como `on`.

A permissão para as ações `start` e `stop` nos objetos `cpu` deve estar ativada no arquivo de segurança para executar este comando.

O estado de correlação das instâncias da regra de correlação pendentes é perdido sempre que o servidor é desligado ou migrado. Se o armazenamento em cache dos eventos recebidos estiver ativado no arquivo de configuração do listener EIF, os eventos armazenados em cache serão perdidos após o processador de eventos ser comutado.

Importante:
- Antes de executar esse comando, execute **`planman deploy`** por precaução. Faça isso para se certificar que as alterações ou inclusões mais recentes nas regras de evento ativas foram implementadas antes do processador de evento ser comutado e, dessa forma, evitar o risco que, por causa de uma incompatibilidade de tempo, as atualizações mais recentes (enviadas automaticamente com base na configuração da opção global **`deploymentFrequency`**) sejam recebidas pelo processador de eventos antigo, em vez do novo.

- O principal e os backups principais designados para executar o processador de eventos devem ter seus relógios sempre sincronizados para evitar inconsistências no cálculo do intervalo de tempo de execução das regras de evento. Na realidade, se o processador de eventos for comutado para um computador não sincronizado, as ações de tempo limite no processo de acionamento poderão sofrer atrasos inesperados. Utilize um servidor Network Time Protocol (NTP) para manter todos os relógios sincronizados.

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Ambiente** > **Monitorar Estações de Trabalho**.
2. Selecione uma tarefa para monitorar estações de trabalho.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. Na tabela que contém a lista de estações de trabalho, selecione uma estação de trabalho e clique em **Mais Ações** > **Tornar o Processador de Evento**.

switchmgr

Alterna o gerenciamento de domínio do gerenciador de domínio atual para um gerenciador de domínio de backup.

Você deve ter acesso ***start*** e ***stop*** ao gerenciador de domínio de backup.

O comando **switchmgr** deve ser utilizado somente como parte de procedimentos específicos para comutar recursos de gerenciamento de domínio de um gerenciador de domínio para seu gerenciador de domínio de backup permanente ou temporariamente. Para obter informações sobre estes procedimentos, consulte o *Tivoli Workload Scheduler: Administration Guide*.

Sintaxe

{**switchmgr** | **switchm**} *domain***;***newmgr*

Argumentos

domínio
Especifica o domínio no qual você deseja alternar gerenciadores.

newmgr
Especifica o nome do novo gerenciador de domínio. Deve ser uma estação de trabalho no mesmo domínio e deve ser definida antecipadamente como um agente tolerante a falhas com Resolver Dependências e Status Completo ativado.

Comentários

O comando pára uma determinada estação de trabalho e a reinicia como o gerenciador de domínio. Todas as estações de trabalho membros do domínio são informadas da comutação e o gerenciador de domínio antigo é convertido em um agente tolerante a falhas no domínio.

Na próxima vez que `JnextPlan` for executado no gerenciador de domínio antigo, o domínio age como se outro comando **switchmgr** tivesse sido executado e o antigo gerenciador de domínio retoma automaticamente as responsabilidades do gerenciamento do domínio.

Agentes tolerantes a falhas definidos com `securitylevel = on` podem falhar em usar a porta SSL para conectar ao novo gerenciador de domínio principal após o comando **switchmgr** ser executado. Nesse caso, execute um dos seguintes para permitir que o agente seja iniciado corretamente:

- Desconecte e, em seguida, conecte o agente ao novo gerenciador de domínio principal.
- Use a opção `securitylevel = force` no agente.

Exemplos

Para alternar o gerenciador de domínio para a estação de trabalho`orca` no domínio`masterdm`, execute o seguinte comando:

```
switchmgr masterdm;orca
```

Para alternar o gerenciador de domínio para a estação de trabalho`ruby` no domínio`bldg2`, execute o seguinte comando:

```
switchmgr bldg2;ruby
```

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Ambiente** > **Monitorar Estações de Trabalho**.
2. Selecione uma tarefa para monitorar estações de trabalho.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. Na tabela que contém a lista de estações de trabalho, selecione uma estação de trabalho e clique em **Mais Ações** > **Tornar o Gerenciador de Domínio Principal**.

system command

Executa um comando do sistema.

Sintaxe

[**:** | **!**] *system-command*

Argumentos

system-command
Especifica qualquer comando de sistema válido. O prefixo (: ou !) será necessário apenas quando um nome de comando for grafado da mesma forma que um comando do **conman**.

Exemplos

Para executar um comando **ps** no UNIX, execute o seguinte comando:

```
ps -ef
```

Para executar um comando **dir** no Windows, execute o seguinte comando:

```
dir \bin
```

tellop

Envia uma mensagem ao console Tivoli Workload Scheduler.

Sintaxe

{**tellop** | **to**} [*text*]

Argumentos

text Especifica o texto da mensagem. A mensagem pode conter até 900 caracteres.

Comentários

Se **tellop** for emitido no gerenciador de domínio principal, a mensagem será enviada a todas as estações de trabalho ligadas. Se emitido em um gerenciador de domínio, a mensagem será enviada a todos os agentes ligados em seu domínio e domínios subordinados. Se emitido em uma estação de trabalho diferente de um gerenciador de domínio, a mensagem será enviada somente a seu gerenciador de domínio, se este estiver ligado. A mensagem será exibida somente se o nível da mensagem de console for maior que zero. Consulte “console” na página 373.

Se **tellop** for digitado sozinho, ele solicitará o texto da mensagem. No prompt, digite cada linha e pressione a tecla Return. No final da mensagem, digite duas barras (//) ou um ponto (.)e pressione a tecla Return. É possível utilizar a seqüência de mudança de linha (\n) para formatar as mensagens. Digitar **Control+c** a qualquer momento, sairá do comando **tellop** sem enviar a mensagem.

Exemplos

Para enviar uma mensagem, execute o seguinte comando:

```
tellop TWS will be stopped at\n4:30 for 15 minutes.
```

Para solicitar texto antes de enviar uma mensagem, execute o seguinte comando:

```
to
TELLOP>**********************************
TELLOP>* TWS will be stopped at  *
TELLOP>*  4:30 para 15 minutos.        *
TELLOP>**********************************
TELLOP>//
```

unlink

Fecha links de comunicação entre estações de trabalho.

Você deve ter acesso ***unlink*** à estação de trabalho de destino.

Sintaxe

unlink [*domain***!**]*workstation*
[**;noask**]

Argumentos

domínio
Especifica o nome do domínio no qual os links devem ser fechados. Não é

necessário especificar o nome de domínio de uma estação de trabalho no domínio mestre. Caracteres curinga são permitidos.

Nota: Especifique sempre o nome de domínio quando desvincular uma estação de trabalho que não esteja no domínio mestre.

Este argumento é útil quando se está desligando mais de uma estação de trabalho em um domínio. Por exemplo, para desligar todos os agentes no domínio`stlouis`, utilize o seguinte comando:

```
unlink stlouis!@
```

Se você não especificar *domain* e *workstation* incluir caracteres curinga, o domínio padrão será aquele no qual o **conman** está sendo executado.

estação de trabalho
Especifica o nome da estação de trabalho a ser desligada. Caracteres curinga são permitidos.

Esse comando não é suportado em estações de trabalho de mecanismo remoto.

noask Especifica para não solicitar confirmação antes da tomada de ação em cada estação de trabalho qualificada.

Comentários

Supondo-se que um usuário tenha acesso **unlink** a estações de trabalho sendo desligadas, as seguintes regras serão aplicadas:

- Um usuário executando o**conman** no gerenciador de domínio principal pode desvincular qualquer estação de trabalho na rede.
- Um usuário executando o**conman** em um gerenciador de domínio diferente do principal pode desvincular qualquer estação de trabalho em seu próprio domínio e em domínios subordinados. O usuário não pode desligar estações de trabalho em domínios ponto a ponto.
- Um usuário executando o**conman** em um agente pode desvincular qualquer estação de trabalho em seu domínio local, contanto que a estação de trabalho seja um gerenciador de domínio ou um host. Um agente de período no mesmo domínio não pode ser desvinculado.

Para obter informações adicionais, consulte “link” na página 386.

Exemplos

A Figura 27 na página 468 e a Tabela 71 na página 468 mostram os links fechados pelos comandos **unlink** executados pelos usuários em vários locais da rede.

DM*n* são gerenciadores de domínio e **A***nn* são agentes.

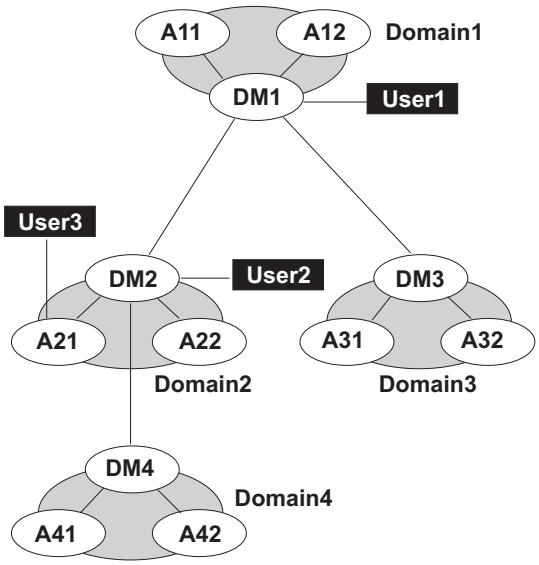


Figura 27. Estações de Trabalho de Rede Desvinculadas

Tabela 71. Estações de Trabalho Desvinculadas

Comando	Fechado pelo Usuário1	Fechado pelo Usuário2	Fechado pelo Usuário3
unlink @!@	Todos os links são fechados	DM1-DM2 DM2-A21 DM2-A22 DM2-DM4 DM4-A41 DM4-A42	DM2-A21
unlink @	DM1-A11 DM1-A12 DM1-DM2 DM1-DM3	DM1-DM2 DM2-A21 DM2-A22 DM2-DM4	DM2-A21
unlink DOMAIN3!@	DM3-A31 DM3-A32	Não permitido	Não permitido
unlink DOMAIN4!@	DM4-A41 DM4-A42	DM4-A41 DM4-A42	Não permitido
unlink DM2	DM1-DM2	Não aplicável	DM2-A21
unlink A42	DM4-A42	DM4-A42	Não permitido
unlink A31	DM3-A31	Não permitido	Não permitido

Consulte também

No Dynamic Workload Console, é possível executar a mesma tarefa da seguinte forma:

1. Na barra de navegação, clique em **Status e Funcionamento do Sistema** > **Monitoramento de Ambiente** > **Monitorar Estações de Trabalho**.
2. Selecione uma tarefa para monitorar estações de trabalho.
3. Escolha um nome de mecanismo, ou especifique propriedades da conexão, e clique em **OK**.
4. Na tabela contendo a lista de estações de trabalho, selecione uma estação de trabalho e clique em **Mais Ações** > **Desvincular...**

version

Exibe o banner do programa **conman**, inclusive da versão até o nível de fix pack instalado.

Sintaxe

{**version** | **v**}

Exemplos

Para exibir o banner do programa **conman**, execute o seguinte comando:

```
%version
```

A saída é semelhante a esta:

```
Tivoli Workload Scheduler (UNIX)/CONMAN 9.1.0.00 (20130516)
Licensed Materials  Property of IBM*
5698-WSH
(C) Copyright IBM Corp 1998, 2013 All rights reserved.
* Trademark of International Business Machines
Installed for user "twsuser".
Locale LANG set to the following: "en"
Scheduled for (Exp) 05/20/13 (#8) on LB001542_MASTER.Batchman LIVES.
Limit:55,Fence:0,Audit Level:0
```

Capítulo 12. Ativando recursos de planejamento dinâmico em seu ambiente

Esta seção explica como é possível ativar as capacidades de planejamento dinâmico em seu ambiente para planejar tarefas existentes do Tivoli Workload Scheduler e do tipos de tarefas com opções avançadas, tanto os fornecidos com o produto, como os tipos adicionais implementados através dos plug-ins customizados.

As capacidades dinâmicas ajudam você a manter políticas de negócios e assegurar os acordos de nível de serviço desta maneira:

- Descobrindo automaticamente os recursos de ambiente de planejamento
- Correspondendo tarefas automaticamente aos recursos disponíveis
- Controlando e otimizando o uso de recursos
- Seguindo automaticamente as mudanças de recurso
- Solicitando recursos adicionais quando necessário

É possível ativar recursos dinâmicos em seu ambiente, definindo um conjunto de tipos de estação de trabalho:

Agente dinâmico
U ma estação de trabalho que gerencia vários tipos de tarefas, por exemplo, tarefas de banco de dados ou de FTP específicas, além dos tipos de tarefas existentes. Essa estação de trabalho é definida automaticamente e registrada no banco de dados do Tivoli Workload Scheduler quando o agente dinâmico é instalado. É possível agrupar agentes dinâmicos em conjuntos e conjuntos dinâmicos.

Em uma rede simples, os agentes dinâmicos se conectam diretamente a seu gerenciador de domínio principal ou por meio de um gerenciador de domínio dinâmico. Em topologias de rede mais complexas, em que o gerenciador de domínio principal ou o gerenciador de domínio dinâmico não possa se comunicar diretamente com o agente dinâmico, é possível configurar os agentes dinâmicos para usarem um gateway local ou remoto.

Conjunto
Uma estação de trabalho que agrupa um conjunto de agentes dinâmicos com características similares de hardware ou software à qual serão enviadas tarefas. O Tivoli Workload Scheduler balanceia as tarefas entre os agentes dinâmicos do conjunto e redesigna tarefas automaticamente a agentes dinâmicos disponíveis se um agente dinâmico não estiver mais disponível. Para criar um conjunto de agentes dinâmicos em seu ambiente do Tivoli Workload Scheduler, defina uma estação de trabalho do tipo **pool** hospedada pela workload broker workstation, em seguida, selecione os agentes dinâmicos que deseja incluir no conjunto. É possível definir o conjunto usando o Dynamic Workload Console ou o comando **composer**.

Conjunto dinâmico
Uma estação de trabalho que agrupa um conjunto de agentes dinâmicos, que é dinamicamente definida com base nos requisitos de recursos especificados por você e hospedada pela workload broker workstation. Por exemplo, se precisar de uma estação de trabalho com um uso de CPU baixo e o Windows instalado para executar sua tarefa, você especifica esses requisitos usando o Dynamic Workload Console ou o comando **composer**. Ao salvar o conjunto de requisitos, uma nova estação de trabalho é criada

© Copyright IBM Corp. 1999, 2014

automaticamente no banco de dados do Tivoli Workload Scheduler. Essa estação de trabalho é hospedada pela workload broker workstation. Esta estação de trabalho mapeia todo o agentes dinâmicos em seu ambiente que atende aos requisitos especificados. O conjunto resultante é atualizado dinamicamente sempre que um novo agente dinâmico adequado for disponibilizado. As tarefas são executadas na primeira estação de trabalho do conjunto dinâmico, a qual atende a todos os requisitos. Tarefas planejadas nessa estação de trabalho herdam automaticamente os requisitos definidos para a estação de trabalho.

Para obter informações sobre como criar conjuntos e conjuntos dinâmicos usando o Dynamic Workload Console, consulte

a seção sobre criar um conjunto de agentes no *Guia do Usuário do Tivoli Dynamic Workload Console*. Para obter mais informações sobre como criar conjuntos e conjuntos dinâmicos usando o comando **composer**, consulte o *Guia do Usuário e Referência*, *S517-7781*.

Os agentes dinâmicos, conjuntos e conjuntos dinâmicos alavancam a funcionalidade dinâmica compilada no Tivoli Workload Scheduler e fornecem a possibilidade em tempo de execução para associar dinamicamente sua carga de trabalho submetida (ou parte dela) aos melhores recursos disponíveis. É possível ativar os recursos de planejamento dinâmico em estações de trabalho no momento da instalação. Para obter mais informações sobre a instalação de agentes dinâmicos, consulte a seção sobre a instalação de um novo agente no *Guia de Planejamento e Instalação*, *S517-7780*.

É possível usar agentes dinâmicos, conjuntos e conjuntos dinâmicos para programar tipos de tarefas com opções avançadas. Os tipos de tarefas com opções avançadas são fornecidos com o produto e os tipos adicionais são implementados por meio dos plug-ins customizados. Ambos os tipos de tarefa são executados em agentes dinâmicos, conjuntos e conjuntos dinâmicos. Para obter informações adicionais sobre como programar tipos de tarefas com opções avançadas, consulte "Criando o tipos de tarefas com opções avançadas" na página 474. Para obter mais informações sobre como criar plug-ins customizados, consulte *Extending Tivoli Workload Automation*, *SC14-7623*.

É possível também usar agentes dinâmicos, conjuntos e conjuntos dinâmicos para executar as tarefas criadas para os tipos de estação de trabalho existentes do Tivoli Workload Scheduler. Para executar essas tarefas nos tipos de estação de trabalho dinâmicos, é necessário apenas alterar a especificação da estação de trabalho onde deseja executar a tarefa. Para obter informações adicionais sobre como programar trabalhos existentes do Tivoli Workload Scheduler, consulte "Incluir Capacidades Dinâmicas em Tarefas do Tivoli Workload Scheduler Existentes" na página 487.

Se quiser usar o recurso dinâmico ao planejar os tipos de tarefas com opções avançadas, você planeja eles em conjuntos e conjuntos dinâmicos, que designam a tarefa dinamicamente ao melhor recurso disponível. Se estiver interessado somente na definição dos tipos de tarefas com opções avançadas, sem usar o recurso de planejamento dinâmico, você planeja essas tarefas em um agente dinâmico específico, no qual a tarefa é executada estaticamente.

Vantagens do tipos de tarefas com opções avançadas

Esta seção descreve as vantagens que podem ser obtidas implementando-se o tipos de tarefas com opções avançadas, tanto os tipos fornecidos com o produto quanto os adicionais implementados através dos plug-ins customizados, e planejando-os no agentes dinâmicos, conjuntos e conjuntos dinâmicos.

Enquanto a tarefa padrão do Tivoli Workload Scheduler é um script ou comando genérico, você pode definir tarefas para executar tarefas específicas como banco de dados, transferência de arquivos, Java e operações de serviço da Web, usando os tipos de tarefas com opções avançadas disponíveis no Dynamic Workload Console ou no comando composer. É possível definir os tipos de tarefas sem ter qualificações específicas sobre os aplicativos nos quais a tarefa é executada.

Os tipos de tarefas com opções avançadas incluem aquelas fornecidas com o produto e os tipos adicionais implementados através dos plug-ins customizados

Os seguintes tipos de tarefas com opções avançadas estão disponíveis

Tarefas de Transferência de Arquivos
Transferir arquivos para e/ou de um servidor acessível usando FTP, SSH ou outros protocolos.

JCL Execute uma tarefa JCL, por referência ou por definição. Se você definir uma tarefa por referência, fornecerá uma referência à tarefa que deseja enviar, sem precisar gravar ou importar a tarefa inteira para a JCL. Se você definir uma tarefa por definição, fornecerá uma definição JCL para ser enviada. Esse tipo de tarefa é executada apenas no Tivoli Workload Scheduler distribuído - Agent for z/OS.

Tarefas de Serviços da Web
Chamar um serviço da Web.

Tarefas de Banco de Dados
Executar consultas, instruções SQL e tarefas em vários bancos de dados, incluindo os bancos de dados customizados. Também é possível criar e executar procedimentos armazenados nos bancos de dados DB2, Oracle e MSSQL.

Tarefas Executáveis
Executar um script ou comando com opções avançadas, como redirecionar entrada e saída padrão para um arquivo.

tarefas de Java
Executar uma classe Java

Tarefas do MSSQL
Execute uma tarefa do Microsoft SQL.

Tarefas de método de acesso
Estendem funções de planejamento de tarefa do Tivoli Workload Scheduler para outros sistemas e aplicativos usando métodos de acesso. Os métodos de acesso se comunicam com o sistema externo para iniciar a tarefa e retornar o status da tarefa. Os seguintes métodos de acesso estão disponíveis:

- Oracle E-Business Suite
- PeopleSoft
- SAP
- MVS

- Métodos customizados

tarefas do J2EE
Permitir que aplicativos Java na mesma rede enviem e recebam mensagens para/de um destino JMS.

tarefas do IBM i
Execute um comando nos sistemas IBM i.

Além de configurar o tipos de tarefas com opções avançadas usando o Dynamic Workload Console ou o comando **composer**, você poderá usar os arquivos de configuração relacionados. Para obter informações adicionais, consulte a seção sobre configuração para planejar os tipos de tarefas com opções avançadas no *Administration Guide*, *SC23-9113*.

Para obter mais informações sobre o procedimento para definir o tipos de tarefas com opções avançadas, consulte as seções sobre as etapas de pré-requisito para criar tipos de tarefas com opções avançadas e sobre criar definições de tarefa no *Guia do Usuário do Tivoli Dynamic Workload Console*.

Para obter informações sobre como criar tarefas usando o comando **composer**, consulte a seção sobre a definição de tarefa no *Guia do Usuário e Referência*, *S517-7781*.

Além disso, é possível criar plug-ins customizados para implementar seus próprios tipos de tarefas com opções avançadas para aplicativos não suportados pelo Tivoli Workload Scheduler. Para obter mais informações sobre como criar plug-ins customizados, consulte *Extending Tivoli Workload Automation*, *SC14-7623*.

Você executa os tipos de tarefas com opções avançadas, aqueles fornecidos com o produto e os tipos adicionais implementados através dos plug-ins customizados somente nos agentes dinâmicos, conjuntos e conjuntos dinâmicos.

Criando o tipos de tarefas com opções avançadas

Esta seção explica como criar um tipo de tarefa específica usando os tipos de tarefas com opções avançadas fornecidos com o Dynamic Workload Console.

É possível definir facilmente os tipos de tarefas com opções avançadas, sem ter qualificações específicas sobre os aplicações nos quais a tarefa é executada. É possível planejar esses tipos de tarefas em agentes dinâmicos, conjuntos e conjuntos dinâmicos. O procedimento a seguir descreve como criar uma tarefa usando o Dynamic Workload Console. O procedimento para criar o tipos de tarefas com opções avançadas é igual, mas cada tipo de tarefa contém opções específicas à tarefa. Para obter informações adicionais sobre cada tipo de tarefa, consulte a Ajuda Online do Dynamic Workload Console.

Para criar uma tarefa usando o Dynamic Workload Console, faça o seguinte:

1. Na barra de ferramentas de navegação, clique em **Administração** > **Design de Carga de Trabalho** > **Gerenciar Definições de Carga de Trabalho**
2. Especifique um nome do mecanismo, distribuído ou z/OS. O Workload Designer é aberto. Os tipos de tarefas e as características variam, dependendo de se você selecionar um mecanismo distribuído ou um z/OS.
3. Na área de janela Lista de Trabalho, selecione **Novo > Definição de Tarefa**.

4. Selecione a categoria e o tipo de tarefa que deseja criar.
5. No painel de propriedades, especifique os atributos para a definição de tarefa que você está criando. Para obter todos os detalhes sobre opções e campos disponíveis, consulte a ajuda online clicando em "**?**" no canto superior direito.
6. Clique em **Salvar** para salvar a definição de tarefa no banco de dados.

Para obter mais informações sobre todos os tipos de tarefa e categorias disponíveis, consulte a seguinte tabela:

Tabela 72. Tipos de Tarefa

Categoria	Tipo de Tarefa	Descrição	tipos de tarefa com opções avançadas
Nativo	Windows	Tarefas executadas nos sistemas operacionais Windows.	NÃO
	UNIX	Tarefas executadas nas plataformas UNIX.	NÃO
	Outro	Tarefas que executam em agentes estendidos. Consulte o *Guia do Usuário do Tivoli Workload Scheduler para Aplicativos* para obter informações sobre tipos de tarefas customizadas para aplicativos adquiridos por fornecedor suportados. Tarefas executadas nos agentes limitados tolerantes a falhas para o IBM i.	NÃO
	Executável	Tarefas que executam script ou comando com opções avançadas, como redirecionamento de entrada padrão e saída padrão para um arquivo.	SIM
	Comando Remoto	São requeridas tarefas que podem ser executadas em computadores remotos e sem instalação do agente. **Nota:** Em sistemas z/OS, ela pode ser criada usando apenas o Dynamic Workload Console	SIM
	z/OS	Tarefas que executam o comando que você especificou na tarefa JCL em um sistema JCL.	SIM
	IBM i	Tarefas que executam um comando nos sistemas IBM i.	SIM
ERP	Tarefa do SAP em Estações de Trabalho XA	Tarefas que executam em um agente estendido SAP. Isso inclui os três tipos de definições de tarefa R/3 do SAP: • Tarefa R/3 padrão • Tarefa da cadeia do processo BW • Tarefa InfoPackage BW	NÃO
	Tarefa do SAP em Estações de Trabalho Dinâmicas	Tarefas que executam nas estações de trabalho do agente dinâmico, conjuntos, conjuntos dinâmicos e agentes z-centric. Os seguintes tipos de definição de tarefa do SAP estão disponíveis: • Tarefa R/3 padrão • Tarefa da cadeia do processo BW • Tarefa InfoPackage BW	NÃO
	Método de Acesso	Tarefas que estendem as funções de planejamento de tarefas do Tivoli Workload Scheduler para outros sistemas e aplicativos usando métodos de acesso. Os métodos de acesso se comunicam com o sistema externo para iniciar a tarefa e retornar o status da tarefa. Os seguintes métodos de acesso estão disponíveis: • Oracle E-Business Suite • PeopleSoft • SAP • MVS • Métodos customizados	NÃO
	Canal de PI	Tarefas que executam tarefas do Canal de Integração de Processo (PI) SAP para controlar os canais de comunicação entre o Integrador de Processo e um sistema SAP R/3 de backend. **Nota:** Fornecido com o Tivoli Workload Scheduler for Applications 8.6 ou mais recente e disponível apenas se especificamente instalado.	SIM
Nuvem	Broker de Carga de Trabalho	Tarefas que gerenciam o ciclo de vida de uma tarefa do broker de carga de trabalho. *Tivoli Workload Scheduler Scheduling Workload Dynamically* para informações sobre como usar o broker de carga de trabalho dinâmica.	NÃO
	Provisioning	Tarefas que expandem computadores físicos, máquinas virtuais e ambientes de nuvem privados e públicos criando um ambiente on-demand. Este tipo de tarefa se integra ao IBM SmartCloud Provisioning. **Nota:** Em sistemas z/OS, ela pode ser criada usando apenas o Dynamic Workload Console	SIM

Tabela 72. Tipos de Tarefa (continuação)

Categoria	Tipo de Tarefa	Descrição	tipos de tarefa com opções avançadas
Transferência de Arquivos e Coordenação	Transferência de Arquivos	Tarefas que executam programa para transferência de arquivos para e a partir de um servidor que pode ser acessado usando FTP, SSH ou outros protocolos.	SIM
	Sombra Distribuída	Tarefas que executam localmente e mapeiam outras tarefas executando em ambiente distribuído do Tivoli Workload Scheduler remoto.	NÃO
	z/OS de sombra	Tarefas executadas localmente e mapeiam outras tarefas em execução no Tivoli Workload Scheduler remoto para o ambiente do z/OS.	NÃO
Banco de Dados e Integrações	Banco de dados	Tarefas que executam consultas, instruções SQL e tarefas em um número de bancos de dados, incluindo banco de dados customizados. É possível criar e executar também os procedimentos armazenados nos bancos de dados DB2, Oracle e Microsoft SQL Server.	SIM
	Serviços da Web	Tarefas que executam um serviço da web.	SIM
	MS SQL	Tarefas que executam uma tarefa do Microsoft SQL Server.	SIM
	J2EE	Tarefas que permitem que os aplicativos Java na mesma rede enviem e recebam mensagens de/para um destino JMS.	SIM
	Java	Tarefas que executam uma classe Java.	SIM
Analítica de Negócios	Relatórios Cognos	Tarefas que executam relatórios do IBM Cognos, relatórios interativos, consulta e visualizações de relatório. **Nota:** Fornecido com o Tivoli Workload Scheduler for Applications 8.6 ou mais recente e disponível apenas se especificamente instalado.	SIM
	InfoSphere DataStage	Tarefas que executam tarefas do IBM InfoSphere DataStage. **Nota:** Fornecido com o Tivoli Workload Scheduler for Applications 8.6 ou mais recente e disponível apenas se especificamente instalado.	SIM
OSLC	OSLC Automation	Tarefas que chamam qualquer provedor OSLC que está implementando o OSLC Automation Specification. Os recursos de automação definem planos de automação, solicitações de automação e resultados de automação do ciclo de vida de desenvolvimento, teste e implementação de software.	SIM
	OSLCProvisioning	Tarefas que chamam qualquer provedor OSLC que está implementando o OSLC Provisioning Specification. Os recursos de fornecimento definem os planos de fornecimento, solicitações de fornecimento e resultados de fornecimento do desenvolvimento de software, testes e ciclo de vida de implementação.	SIM

Códigos de Retorno

A seguir há uma lista dos códigos de retorno para o tipos de tarefas com opções avançadas

```
Tarefas do banco de dados:
RC = 0 -> A tarefa foi concluída com sucesso
RC = -1 -> Uma instrução SQL foi executada com um código de saída
diferente de 1
RC = -2 -> Erro de Tarefa MSSQL
RC = -3 -> Uma instrução SQL não foi executada por causa de um
erro na instrução

Tarefas de transferência de arquivos:
RC = 0 -> A transferência de arquivos foi concluída com êxito

RC = -1 -> A transferência de arquivos não foi executada. A tarefa falha com o seguinte
código de erro: AWKFTE007E

Explicação: Ocorreu um erro durante a operação de transferência de arquivos

Motivos possíveis: Arquivo remoto não localizado ou permissão negada

RC = -2 -> A transferência de arquivos não foi executada. A tarefa falha com o seguinte
código de erro: AWKFTE020E

Explicação: Apenas para protocolos SSH ou Windows. Foi retornado um erro
durante a tentativa de converter a página de códigos
```

```
Motivos possíveis: Para protocolos SSH ou Windows, a página de códigos é
automaticamente detectada e convertida. Nesse caso, há
um erro na
página de códigos do arquivo a ser transferida, que não é compatível com a
página de códigos do sistema local

RC = -3 -> A transferência de arquivos não foi executada. A tarefa falha com o seguinte
código de erro: AWKFTE015E

Explicação: Ocorreu um erro durante a operação de transferência de arquivos

Motivos possíveis: Arquivo local não localizado

RC = -4 -> A transferência de arquivos é executada com a página de códigos padrão. A tarefa
falha com o código de erro a seguir: AWKFTE023E

Explicação: A conversão de página de códigos especificada não foi executada.
A transferência de arquivos foi executada com páginas de códigos padrões

Motivos possíveis: O página de códigos especificada não é válida

Tarefas IBM i:
Código de retorno = código de retorno do usuário quando recuperado
Código de retorno = 0 -> tarefa concluída com êxito
Código de retorno > -1 -> tarefa concluída sem êxito

Tarefas Java:
RC = 0 -> A tarefa foi concluída com sucesso
RC = -1 -> O aplicativo Java ativado pela tarefa falhou devido a
uma exceção

Tarefas de serviços da web:
RC = 0 -> A tarefa foi concluída com sucesso
RC = -1 -> O nome do host do servidor contido na URL de Serviço da
Web é desconhecido
RC = -2 -> Erro de inovação de Serviço da Web
```

Quando o código de retorno do usuário é recuperado, o IBM i Agent Monitor designa uma prioridade a ele.

Definindo Variáveis e Senhas para Resolução Local em Agentes Dinâmicos

Para tipos de tarefa com opções avançadas você tem a possibilidade de deixar que variáveis e senhas sejam definidas e resolvidas localmente nos agentes dinâmicos (incluindo conjuntos e conjuntos dinâmicos).

Isso é particularmente útil no caso de senhas porque você não é solicitado a especificá-las na definição de tarefa. A vantagem é que, se a senha tiver de mudar, você não modifica a definição de tarefa, mas a muda com o comando param localmente nos agentes (ou nos agentes do conjunto) que executam ou podem executar a tarefa. Se a tarefa for enviada para um conjunto ou conjunto dinâmico, será possível copiar o arquivo com as definições de variável para todos os agentes que participam nesse conjunto, para que as variáveis sejam resolvidas localmente onde quer que a tarefa seja executada.

Esse recurso não é restrito às estações de trabalho do Windows sozinhas. Também é possível usá-lo em UNIX, desde que você o aplique em tipos de tarefa com opções avançadas.

Para definir uma variável ou uma senha localmente em um agente dinâmico, use o comando do utilitário param. Esse comando tem o poder para criar, excluir e listar variáveis locais em agentes dinâmicos. Consulte os detalhes sobre esse comando para aprender como usá-lo.

Especificando Variáveis Locais e Senhas nas Definições de Tarefa

Após definir uma variável e seu valor com o comando `param`, para incluí-la dentro de uma definição de tarefa para que ela seja resolvida localmente no agente no tempo de execução, use a seguinte sintaxe:

```
${agent:variable_name}
```

Após definir uma senha com o comando `param`, para incluí-la dentro de uma definição de tarefa para que ela seja resolvida localmente no agente no tempo de execução, use a seguinte sintaxe:

```
${agent:password.user_name}
```

É possível aninhar variáveis dentro de senhas. Se você usou uma variável para definir um usuário, insira a senha como a seguir dentro da definição de tarefa:

```
${agent:password.${agent:variable_name}}
```

em que *variable_name* foi definida previamente como tendo valor *user_name* com o comando `param`.

Exemplo

Um administrador do Tivoli Workload Scheduler precisa definir uma tarefa de transferência de arquivos que efetue o download de um arquivo de um servidor remoto para um dos agentes dinâmicos que compõem um conjunto de agentes. O administrador deseja parametrizar na definição de tarefa:

- O nome com o qual o arquivo remoto será salvo localmente
- O nome do usuário remoto e sua senha

O administrador continua da seguinte maneira em um dos agentes:

1. Define uma variável chamada `localfile`. A variável recebe um valor igual a `./npp.5.1.1.Installer.DW.exe` e é criada em um novo arquivo de variáveis chamado `FTPvars` (sem seção). O comando para fazer isso é:

   ```
   E:\IBM\TWA\TWS\CLI\bin>param -c FTPvars..localfile
   ./npp.5.1.1.Installer.DW.exe
   ```

2. Define uma variável chamada `remoteUser`. A variável recebe um valor igual a `FTPuser` e é criada no arquivo `FTPvars` (sem seção). O comando para fazer isso é:

   ```
   E:\IBM\TWA\TWS\CLI\bin>param -c FTPvars..remoteUser FTPuser
   ```

3. Define a senha para `FTPuser`. O valor da senha é `tdwb8nxt` e é criado na seção `password` do arquivo `FTPvars`. O comando para fazer isso é:

   ```
   E:\IBM\TWA\TWS\CLI\bin>param -c FTPvars.password.FTPuser tdwb8nxt
   ```

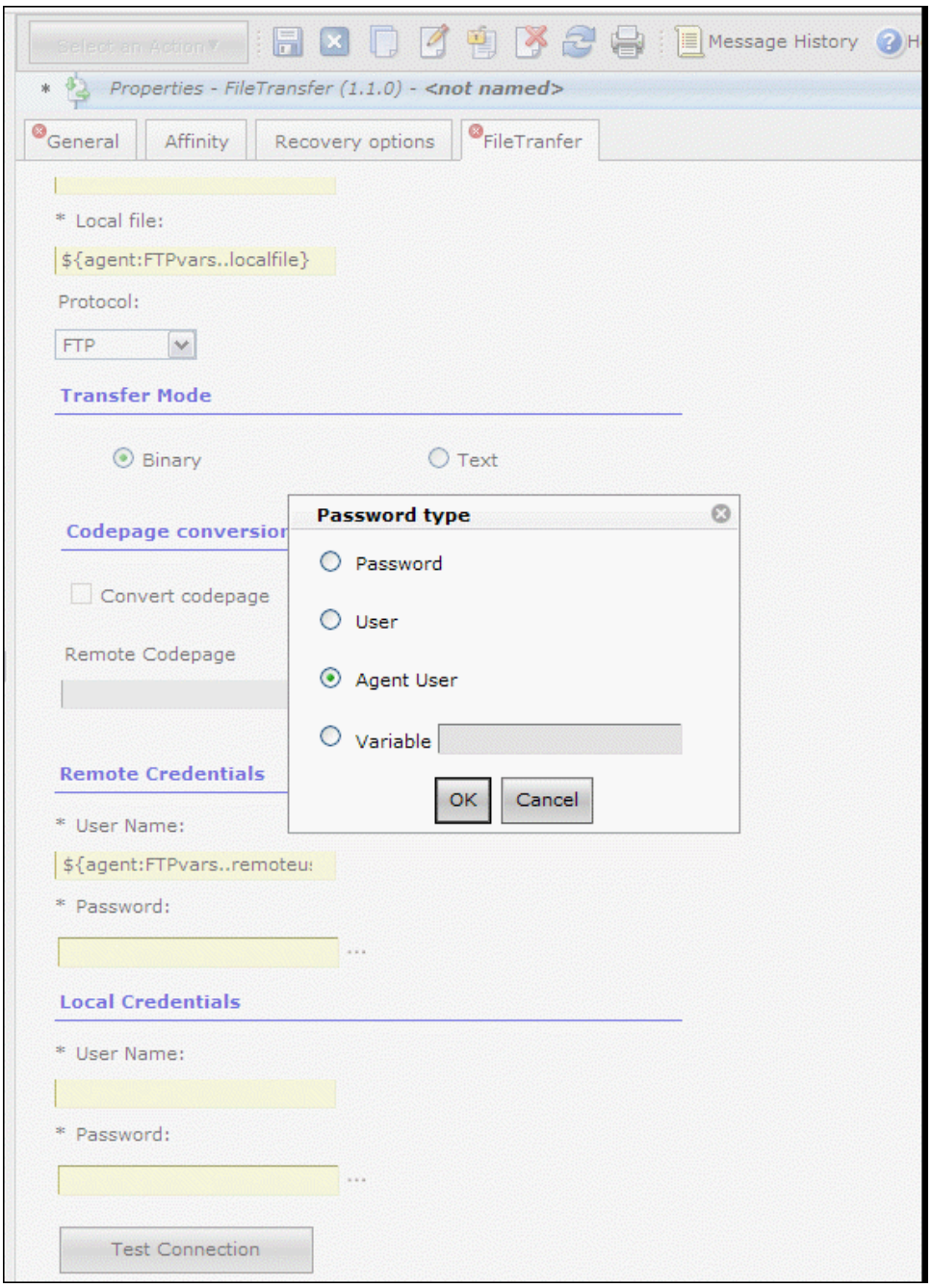
4. Com um editor de texto abre o arquivo `E:\IBM\TWA\TWS\ITA\cpa\config\jm_variables_files\FTPvars` e verifica seu conteúdo:

   ```
   localfile = ./npp.5.1.1.Installer.DW.exe
   remoteuser = FTPuser

   [password]
   FTPuser = {aes}XMMYMY2zBHvDEDBo5DdZVmwOJao60pX1K6x2HhRcovA=
   ```

5. Copia o arquivo `FTPvars` no caminho *agent_installation_path*`\TWA\TWS\ITA\cpa\config\jm_variables_files>` de cada um dos outros agentes definidos no conjunto.

6. Inicia definindo a nova tarefa de transferência de arquivos no painel do Workload Designer do Dynamic Workload Console. Na janela `FileTransfer`:
   a. Insere `${agent:FTPvars..localfile}` no campo `Arquivo local`.
   b. Insere `${agent:FTPvars..remoteuser}` no campo `Credenciais Remotas →Nome de Usuário .`
   c. Clica no botão `...` ao lado do campo `Credenciais Remotas →Senha`. A janela `Tipo de senha` é exibida e o administrador seleciona `Usuário do Agente`.



   d. Após o administrador clicar no botão `OK` para confirmação na janela pop-up, o campo `Credenciais Remotas →Senha` é preenchido com o valor `${agent:password.${agent:FTPvars..remoteuser}}`.
7. Preenche todos os outros campos para concluir a definição de tarefa.

Quando a tarefa é executada, as entidades e a senha inseridas como variáveis são resolvidas com os valores definidos no arquivo `FTPvars`.

Usando Variáveis em Tarefas do Dynamic Workload Broker

Essa seção explica como incluir variáveis para tarefas que você planeja executar com o Dynamic Workload Broker.

É possível incluir variáveis na sua definição de tarefa. As variáveis são resolvidas no tempo de envio.

As variáveis suportadas são as seguintes:

Tabela 73. Variáveis Tivoli Workload Scheduler suportadas nas definições JSDL.

Variáveis que podem ser inseridas na definição de tarefa do Dynamic Workload Broker	Descrição
tws.host.workstation	Nome da estação de trabalho host.
tws.job.date	Data da tarefa enviada.
tws.job.fqname	Nome completo da tarefa (UNISON_JOB).
tws.job.ia	Horário de chegada da entrada da tarefa.
tws.job.interactive	A tarefa é interativa. Os valores podem ser `verdade` ou `falso`. Aplica-se apenas às tarefas compatíveis com versões anteriores.
tws.job.logon	As credenciais do usuário que executa a tarefa (LOGIN). Aplica-se apenas às tarefas compatíveis com versões anteriores.
tws.job.name	Nome da tarefa enviada.
tws.job.num	UNISON_JOBNUM.
tws.job.priority	Prioridade da tarefa enviada.
tws.job.promoted	A tarefa é promovida. Os valores podem ser `SIM` ou `NÃO`. Para obter informações adicionais sobre promoção para tarefas dinâmicas, consulte "Promovendo Tarefas Programadas em Conjuntos Dinâmicos" na página 486.
tws.job.recnum	Número de registro da tarefa.
tws.job.resourcesForPromoted	Quantidade dos recursos lógicos necessários designados em um conjunto dinâmico para uma tarefa promovida. Os valores podem ser 1 se a tarefa for promovida ou 10 se a tarefa não for promovida. Para obter informações adicionais sobre promoção para tarefas dinâmicas, consulte "Promovendo Tarefas Programadas em Conjuntos Dinâmicos" na página 486.
tws.job.taskstring	Sequência de tarefas da tarefa enviada. Aplica-se apenas às tarefas compatíveis com versões anteriores.
tws.job.workstation	Nome da estação de trabalho na qual a tarefa está definida.
tws.jobstream.id	ID do fluxo de tarefas que inclui a tarefa (UNISON_SCHED_ID).
tws.jobstream.name	Nome do fluxo de tarefas que inclui a tarefa (UNISON_SCHED).
tws.jobstream.workstation	Nome da estação de trabalho na qual o fluxo de tarefas que inclui a tarefa está definido.
tws.master.workstation	Nome do gerenciador de domínio principal (UNISON_MASTER).
tws.plan.date	Data de início do plano de produção (UNISON_SCHED_DATE).
tws.plan.date.epoch	Data de início do plano de produção, em formato de época (UNISON_SCHED_EPOCH).
tws.plan.runnumber	Número de execução do plano de produção (UNISON_RUN).

Passando Variáveis Entre Tarefas na Mesma Instância do Fluxo de Tarefas

Em muitos cenários, a saída de tarefa ou uma propriedade de tarefa da primeira tarefa em um fluxo de tarefas pode ser a entrada para a execução de tarefas sucessivas na mesma instância do fluxo de tarefas. Isto é válido também para o fluxo de tarefas `JOBS`.

No cenário, você tem *JobA* e *JobB* na mesma instância do fluxo de tarefas e *JobA* é um predecessor de *JobB*.*JobA* passa alguns valores de variáveis para *JobB* no tempo de execução.

É possível transmitir as seguintes variáveis do *JobA* para *JobB*:

- *JobA* exporta algumas propriedades e *JobB* faz referência a essas propriedades em sua definição como variáveis em um formato predefinido. No tempo de execução, as variáveis *JobB* são automaticamente resolvidas. As propriedades da tarefa que podem ser exportadas dependem do tipo de tarefa que você está definindo. Consulte o "Transmitindo propriedades da tarefa de uma tarefa para outra na instância do mesmo fluxo de tarefas".
- *JobA* exporta seu valor de saída padrão e *JobB* referencia essa saída padrão como uma variável. No tempo de execução, a variável*JobB* é automaticamente resolvida. Consulte o "Transmitindo uma saída padrão de tarefa de uma tarefa para outra na instância do mesmo fluxo de tarefas" na página 483.
- *Apenas para tarefas executáveis. JobA* exporta seu valor de saída padrão e *JobB* referencia essa saída padrão como seu valor de entrada padrão. Consulte o "Transmitindo uma saída padrão de tarefa de uma tarefa para outra como entrada padrão na instância do mesmo fluxo de tarefas" na página 484.
- *JobA* define alguns valores da variável utilizando o utilitário **jobprop** nos sistemas operacionais UNIX e o utilitário **jobprop.exe** em sistemas operacionais Windows que são instalados no agentes dinâmicos. *JobB* referência esses valores da variável em sua definição. No tempo de execução, as variáveis *JobB* são automaticamente resolvidas. Essa opção é válida somente para tarefas nativas e executáveis. Consulte o "Transmitindo variáveis configuradas usando o **jobprop** de uma tarefa para outra na instância do mesmo fluxo de tarefas" na página 484.

Nota: O fluxo de tarefas USERJOBS que é criado pelos processos Tivoli Workload Scheduler, não suporta a transmissão de variáveis entre tarefas que pertencem a ele.

Transmitindo propriedades da tarefa de uma tarefa para outra na instância do mesmo fluxo de tarefas

As propriedades da tarefa que você pode exportar de uma tarefa dinâmica para uma tarefa sucessiva na mesma instância do fluxo de tarefas dependem do tipo de tarefa que você está definindo. Para incluir uma propriedade da tarefa dentro de outra definição de tarefa sucessora, para que ela seja resolvida localmente no agente no tempo de execução, use a seguinte sintaxe:

```
${job:<JOB_NAME>.<property_name>}
```

em que <*JOB_NAME*> é o valor do nome ou valor de nome de alias da tarefa a partir da qual você está exportando os valores de propriedade e <*property_name*> é a propriedade a que você está se referindo. O valor <*property_name*> não faz distinção entre maiúsculas e minúsculas.

Apenas alguns tipos de tarefa podem transmitir valores da propriedade para outras tarefas sucessoras. Os seguintes tipos de tarefa podem exportar variáveis:

Tarefas de InfoSphere DataStage
O Tabela 74 na página 482 mostra a lista de propriedades que você pode passar de uma tarefa do InfoSphere DataStage para outra e indica o mapeamento entre as propriedades de Informações extras da tarefa e as propriedades que você pode utilizar. Para obter mais informações sobre as tarefas do InfoSphere DataStage, consulte o *IBM Tivoli Workload Scheduler for Applications Guia do Usuário*.

Tarefas sombra
O Tabela 75 na página 483 mostra a lista de propriedades que você pode passar de uma tarefa sombra para outra e indica o mapeamento entre as propriedades de Informações extras da tarefa e as propriedades que você pode utilizar.

Tarefas de OSLC

O Tabela 76 na página 483 mostra a lista de propriedades que você pode passar de uma tarefa do OSLC para outra e indica o mapeamento entre as propriedades de Informações extras da tarefa e as propriedades que você pode utilizar.

Exemplo

O exemplo a seguir demonstra como especificar variáveis em formatos diferentes permite que as variáveis tenham valores diferentes, porque elas são resolvidas em momentos diferentes. Ele também demonstra como as variáveis podem ser transmitidas de tarefa para tarefa em uma instância do fluxo de tarefas. O fluxo de tarefas WIN92MAS_REW#VP_JS_141800058 contém as tarefas JOBA eJOBB. A tarefa executável JOBB faz referência às seguintes propriedades da tarefa sombra JOBA:

- ScheduledTime
- dJobNAme
- dJobStreamName
- dJobStreamWorkstation

As definições do banco de dados:

```
SCHEDULE WIN92MAS_REW#VP_JS_141800058
:
WIN92MAS_REW#JOBA
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:dshadow=
"http://www.ibm.com/xmlns/prod/scheduling/1.0/dshadow" xmlns:
 jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl">
  <jsdl:application name="distributedShadowJob">
    <dshadow:DistributedShadowJob>
      <dshadow:JobStream>VPJS_141800058</dshadow:JobStream>
      <dshadow:Workstation>ncl25133</dshadow:Workstation>
      <dshadow:Job>VP_JOBMON_141800058</dshadow:Job>
      <dshadow:matching>
        <dshadow:previous/>
      </dshadow:matching>
    </dshadow:DistributedShadowJob>
  </jsdl:application>
</jsdl:jobDefinition>
 DESCRIÇÃO "Definição da Tarefa de Amostra para o ambiente DISTRIBUÍDO"
 RECOVERY STOP
NC125133#JOBB
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:XMLSchema=
"http://www.w3.org/2001/XMLSchema" xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl" xmlns:
jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle" XMLSchema:text="resolveVariableTable" name="executable">
  <jsdl:application name="executable">
    <jsdle:executable>
            <jsdle:script>
echo ScheduledTime:${job:JOBA.ScheduledTime}
echo JobName:${job:JOBA.dJobName}
echo JobStreamName:${job:JOBA.dJobStreamName}
echo JobStreamWorkstation:${job:JOBA.dJobStreamWorkstation}
</jsdle:script>
        </jsdle:executable>
  </jsdl:application>
</jsdl:jobDefinition>
 DESCRIPTION "Added by composer."
 RECOVERY STOP
 FOLLOWS JOBA

FIM
```

O Tivoli Workload Scheduler usa as propriedades das tarefas do InfoSphere DataStage no estado em que se encontram. Os valores da propriedade Informações Extras das tarefas do InfoSphere DataStage dependem do código do idioma da estação de trabalho onde o InfoSphere DataStage está instalado. O Tabela 74 mostra os valores da propriedade Informações Extras da tarefa do InfoSphere DataStage para uma estação de trabalho com o código do idioma configurado como en_US.

Tabela 74. Propriedades para tarefas do InfoSphere DataStage

Propriedades da tarefa do InfoSphere DataStage que podem ser passadas para outra definição de tarefa	Propriedades de Informações Extras da tarefa do InfoSphere DataStage
${job:<*JOB_NAME*>.Interim Status}	Status Provisório
${job:<*JOB_NAME*>.Invocation ID}	ID de Chamada

Tabela 74. Propriedades para tarefas do InfoSphere DataStage (continuação)

Propriedades da tarefa do InfoSphere DataStage que podem ser passadas para outra definição de tarefa	Propriedades de Informações Extras da tarefa do InfoSphere DataStage
${job:<*JOB_NAME*>.Invocation List}	Invocation List
${job:<*JOB_NAME*>.Job Control}	Controle de Tarefa
${job:<*JOB_NAME*>.Job Controller}	Controlador de Tarefa
${job:<*JOB_NAME*>.Job Process ID}	ID do Processo de Tarefas
${job:<*JOB_NAME*>.Job Restartable}	Tarefa Reiniciável
${job:<*JOB_NAME*>.Job Start Time}	Horário de Início da Tarefa
${job:<*JOB_NAME*>.Job Status}	Status da Tarefa
${job:<*JOB_NAME*>.Job Wave Number}	Número de Renúncia da Tarefa
${job:<*JOB_NAME*>.Last Run Time}	Horário da Última Execução
${job:<*JOB_NAME*>.User Status}	Status do Usuário

Tabela 75. Propriedades para tarefas sombra

Propriedades da tarefa sombra que podem ser transmitidas para outra definição de tarefa	Propriedades de Informações Extras da tarefa sombra
${job:<*JOB_NAME*>.ScheduledTime}	Horário Planejado da Tarefa Remota
${job:<*JOB_NAME*>.dJobName}	Tarefa Remota
${job:<*JOB_NAME*>.dJobStreamName}	Fluxo de Tarefas Remoto
${job:<*JOB_NAME*>.dJobStreamWorkstation}	Estação de Trabalho do Fluxo de Tarefas Remoto

Tabela 76. Propriedades para tarefas do OSLC

Propriedades da tarefa OSLC que podem ser passadas para outra tarefa	Propriedades de Informações Extras da tarefa do OSLC
${job:<*JOB_NAME*>.RESULT_URI}	RESULT_URI

Transmitindo uma saída padrão de tarefa de uma tarefa para outra na instância do mesmo fluxo de tarefas

Você pode exportar a saída padrão da tarefa a partir de uma tarefa dinâmica para uma tarefa sucessiva na mesma instância do fluxo de tarefas. A variável de saída padrão da tarefa é utilizada no campo `script` da definição de tarefa

Para incluir uma saída padrão dentro de outra definição de tarefa, para que ela seja resolvida localmente no agente no tempo de execução, use a seguinte sintaxe:

```
${job:<JOB_NAME>.stdlist}
```

em que <*JOB_NAME*> é o nome ou o nome alternativo da tarefa a partir da qual você está exportando a saída padrão da tarefa.

Exemplo

Neste exemplo, o fluxo de tarefas `WIN92MAS_REW#VP_JS_141800058` contém `JOBA_ALIAS` que é o alias do `JOBA` e as tarefas do `JOBB`. A tarefa executável `JOBB` faz referência à saída padrão `JOBA_ALIAS`.

As definições do banco de dados:

```
SCHEDULE WIN92MAS_REW#VP_JS_141800058
:
WIN92MAS_REW#JOBA como JOBA_ALIAS
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:dshadow=
"http://www.ibm.com/xmlns/prod/scheduling/1.0/dshadow" xmlns:
 jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl">
  <jsdl:application name="distributedShadowJob">
    <dshadow:DistributedShadowJob>
      <dshadow:JobStream>VPJS_141800058</dshadow:JobStream>
      <dshadow:Workstation>ncl25133</dshadow:Workstation>
      <dshadow:Job>VP_JOBMON_141800058</dshadow:Job>
      <dshadow:matching>
        <dshadow:previous/>
      </dshadow:matching>
    </dshadow:DistributedShadowJob>
  </jsdl:application>
</jsdl:jobDefinition>
 DESCRIÇÃO "Definição de Tarefa de Amostra para o ambiente DISTRIBUÍDO"
```

```
 RECOVERY STOP

NC125133#JOBB
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:XMLSchema="http://www.w3.org/2001/XMLSchema" xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl" xmlns:
jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle" XMLSchema:text="resolveVariableTable" name="executable">
  <jsdl:application name="executable">
    <jsdle:executable>
    <jsdle:script>echo &quot;stdlist: ${job:JOBA_ALIAS.stdlist}&quot;
</jsdle:script>
    </jsdle:executable>
  </jsdl:application>
</jsdl:jobDefinition>
 DESCRIPTION "Added by composer."
 RECOVERY STOP
 FOLLOWS JOBA_ALIAS
FIM
```

Transmitindo uma saída padrão de tarefa de uma tarefa para outra como entrada padrão na instância do mesmo fluxo de tarefas

Você pode exportar a saída padrão da tarefa a partir de uma tarefa dinâmica para uma tarefa executável sucessiva como entrada padrão na mesma instância do fluxo de tarefas. A variável de saída padrão da tarefa é utilizada no campo `input` da definição de tarefa executável

Para incluir uma saída padrão dentro de outra definição de tarefa executável, para que ela seja resolvida localmente no agente no tempo de execução, use a seguinte sintaxe:

```
${job:<JOB_NAME>.stduri}
```

em que *<JOB_NAME>* é o valor do nome ou valor de nome alternativo da tarefa a partir da qual você está exportando a saída padrão da tarefa.

Nota: A passagem da variável *stduri* não é suportada para tarefas sombra.

Exemplo

Neste exemplo, o fluxo de tarefas `NC112019#JS_PROP` contém o `JOBALIAS_A` que é o alias `NC112019#JOBA` e as tarefas`NC112019#JOBB`. A tarefa executável `NC112019#JOBB` faz referência à saída padrão`JOBALIAS_A` como entrada padrão.

As definições do banco de dados:

```
SCHEDULE NC112019#JS_PROP
:
NC112019#JOBA AS JOBALIAS_A
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl" xmlns:
jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle" name="executable">
  <jsdl:application name="executable">
    <jsdle:executable interactive="false" path="ls"/>
  </jsdl:application>
</jsdl:jobDefinition>
 DESCRIÇÃO "Incluído pelo editor no fluxo de tarefas: NC112019#JS_PROP."
 RECOVERY STOP

NC112019#JOBB
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:XMLSchema="http://www.w3.org/2001/XMLSchema" xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl" xmlns:
jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle" XMLSchema:text="resolveVariableTable" name="executable">
  <jsdl:application name="executable">
    <jsdle:executable input="${job:JOBALIAS_A.stduri}" interactive="false" path="cat"/>
  </jsdl:application>
</jsdl:jobDefinition>
 DESCRIPTION "Incluído pelo editor no fluxo de tarefas: WIN92MAS#JS_PROP."
 RECOVERY STOP
 FOLLOWS JOBALIAS_A
FIM
```

Transmitindo variáveis configuradas usando o `jobprop` de uma tarefa para outra na instância do mesmo fluxo de tarefas

Você pode utilizar o utilitário **`jobprop`** instalado no agentes dinâmicos para configurar a variável e seu valor em uma tarefa e transmitir a variável para a tarefa sucessiva na mesma instância do fluxo de tarefas.

Para configurar a variável e seu valor na primeira tarefa, utilize a seguinte sintaxe:

```
jobprop <VAR-NAME> <value>
```

em que <*VAR-NAME*> é a variável que você pode exportar para outra tarefa e `<value>` é o valor designado para o <*VAR-NAME*>. Para obter informações adicionais sobre o utilitário **jobprop**, consulte"jobprop" na página 540.

Para definir as variáveis em outra tarefa, utilize a seguinte sintaxe:

```
${job:<JOB_NAME>.<VAR-NAME>}
```

em que <*JOB_NAME*> é o valor do nome ou o valor do nome alternativo da tarefa a partir da qual você está exportando o valor da variável <*VAR-NAME*>.

Exemplo

Neste exemplo, o fluxo de tarefas `WIN92MAS#JS_PROP` contém as tarefas executáveis`NC125133#JOBA` e `NC125133#JOBB`. A tarefa `NC125133#JOBB` referencia os seguintes valores de variáveis do `NC125133#JOBA` configurados usando o utilitário **jobprop**:

- *VAR1* variável configurada para o valor `value1`.
- *VAR2* variável configurada para o valor `value2`.
- *VAR3* variável configurada para o valor `value3`.
- *VAR4* variável configurada para o valor `value4`.

As definições do banco de dados:

```
SCHEDULE WIN92MAS#JS_PROP
:
NC125133#JOBA
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl" xmlns:
jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle">
  <jsdl:application name="executable">
    <jsdle:executable interactive="false">
    <jsdle:script>#!/bin/sh
. /home/ITAuser/TWA/TWS/tws_env.sh
jobprop VAR1 value1
jobprop VAR2 value2
jobprop VAR3 value3
jobprop VAR4 value4
</jsdle:script>
    </jsdle:executable>
  </jsdl:application>
</jsdl:jobDefinition>
 DESCRIPTION "Sample Job Definition"
 RCCONDSUCC "RC>=0"
 RECOVERY STOP

NC125133#JOBB
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl" xmlns:
jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle">
  <jsdl:application name="executable">
    <jsdle:executable interactive="false">
    <jsdle:script>
echo VAR1=${job:joba.VAR1}
echo VAR2=${job:joba.VAR2}
echo VAR3=${job:joba.VAR3}
echo VAR4=${job:joba.VAR4}
</jsdle:script>
    </jsdle:executable>
  </jsdl:application>
</jsdl:jobDefinition>
 DESCRIPTION "Sample Job Definition"
 RCCONDSUCC "RC>=0"
 RECOVERY STOP
 FOLLOWS JOBA
FIM
```

Definindo Relacionamentos de Afinidade

Relacionamentos de afinidade fazem as tarefas executarem na mesma estação de trabalho. A estação de trabalho na qual a primeira tarefa executa é escolhida dinamicamente pelo Dynamic Workload Broker, e a tarefa ou tarefas relacionadas executam na mesma estação de trabalho. Essa seção se aplica ao tipos de tarefas com opções avançadas e às tarefas do broker de carga de trabalho.

No Dynamic Workload Broker, é possível definir relacionamentos de afinidade entre duas ou mais tarefas quando desejar que elas executem na mesma estação de trabalho. Ao enviar a tarefa do ambiente Tivoli Workload Scheduler, é possível definir a afinidade que será resolvida pelo Dynamic Workload Broker incluindo uma definição de afinidade à seção **Sequência de Tarefas** da tarefa Tivoli Workload Scheduler usando o nome da tarefa Tivoli Workload Scheduler como a seguir:

```
jobName [-var varName=varValue,...,]-twsaffinity jobname=twsJobName
```

em que

twsJobName
: É o nome da instância da tarefa Tivoli Workload Scheduler com a qual você deseja estabelecer um relacionamento de afinidade.

Nota: As tarefas devem pertencer ao mesmo fluxo de tarefas

Promovendo Tarefas Programadas em Conjuntos Dinâmicos

Esta seção explica como promover uma tarefa crítica programada em um conjunto dinâmico. Uma tarefa promovida pode ser executada em um número maior de agentes dinâmicos no conjunto dinâmico do que uma tarefa não promovida. Isso assegura que uma tarefa importante seja executada antes de outras tarefas menos importantes.

Para assegurar que uma tarefa crítica obtenha os recursos necessários e seja processada dentro do prazo, especifique as seguintes variáveis:

tws.job.promoted
: Essa variável indica se a tarefa é promovida. Os valores suportados são **yes** e **no** . A seguir estão os eventos de tarefa que podem ser monitorados:

tws.job.resourcesForPromoted
: Essa variável é determinada na definição do conjunto dinâmico e indica a quantidade dos recursos lógicos designados em um conjunto dinâmico para uma tarefa promovida. Isto pode acontecer se a tarefa não for válida, se a tarefa já foi arquivada ou se o armazenamento de tarefas estiver incorretamente estruturado. A quantidade é indicada com esta notação: **${tws.job.resourcesForPromoted}**.

Quando uma tarefa é planejada no conjunto dinâmico, o valor da variável **tws.job.promoted** na tarefa determina o comportamento do conjunto dinâmico:

- Se o valor da variável **tws.job.promoted** for **NO**, o valor da variável **tws.job.resourcesForPromoted** no conjunto dinâmico será 10, o que significa que poucos recursos correspondem ao requisito.
- Se o valor da variável **tws.job.promoted** for **YES**, o valor da variável **tws.job.resourcesForPromoted** no conjunto dinâmico será 1, o que significa que mais recursos correspondem ao requisito, porque o conjunto dinâmico inclui

estações de trabalho com quantidade de recursos igual ou maior que 1 e não apenas estações de trabalho com valor igual ou maior que 10.

Por exemplo, é possível gravar um script que verifique o valor designado para a variável **tws.job.promoted** e execute diferentes ações dependendo da tarefa ser ou não promovida.

Incluir Capacidades Dinâmicas em Tarefas do Tivoli Workload Scheduler Existentes

Esta seção explica como modificar uma tarefa existente para usar as capacidades dinâmicas com agentes dinâmicos, conjuntos e conjuntos dinâmicos.

É possível modificar as tarefas existentes do Tivoli Workload Scheduler para usar as capacidades dinâmicas fornecidas com agentes dinâmicos, conjuntos e conjuntos dinâmicos. Para modificar uma tarefa existente, proceda da seguinte forma:

1. Identificar a instalação existente do agentes dinâmicos.
2. Opcionalmente, designe os agentes dinâmicos aos conjuntos ou crie conjuntos dinâmicos com base em seus requisitos.
3. Analise as tarefas existentes do Tivoli Workload Scheduler e decida quais delas obterão os melhores resultados ao usar a capacidade dinâmica.
4. Efetue login no Dynamic Workload Console.
5. Na barra de ferramentas de navegação, clique em **Administração** > **Design de Carga de Trabalho** > **Gerenciar Definições de Carga de Trabalho**
6. Especifique um nome do mecanismo, distribuído ou z/OS. O Workload Designer é aberto. Os tipos de tarefas e as características variam, dependendo de se você selecionar um mecanismo distribuído ou um z/OS.
7. No painel Lista de Trabalho, selecione **Novo > Definição de Tarefa > FileTransfer**. O painel Desinstalação é exibido.
8. Preencha os campos para especificar seus critérios de procura e clique em **Procurar**.
9. Selecione uma ou mais tarefas entre os resultados da procura e clique em **Editar**. As propriedades da tarefa são exibidas no painel direito para edição.
10. Na guia **Geral**, clique no botão de procura do campo **Estação de Trabalho**. O painel Desinstalação é exibido.
11. Preencha os campos para especificar seus critérios de procura e clique em **Procurar**.
12. Selecione o agente dinâmico, conjunto ou conjunto dinâmico apropriado e clique em **OK**. A tarefa agora é designada à estação de trabalho especificada e será executada quando planejada.

Um Cenário de Negócios

Esta seção demonstra um cenário de negócios de amostra que descreve as vantagens de tipos de tarefas com opções avançadas e capacidade dinâmica.

Uma empresa de segurança executa uma série de tarefas à noite para salvar os dados processados durante o dia no banco de dados de backup. Eles também precisam reunir todos os dados sobre as transações concluídas durante o dia de todas as estações de trabalho nas filiais da empresa. Eles usam o banco de dados

DB2. Usando os tipos de tarefas com opções avançadas fornecidos no Designer de Carga de Trabalho, eles criam uma tarefa para executar um backup do banco de dados e outra tarefa para extrair os dados da transações diárias. Para executar essas operações, eles usam o novo banco de dados tipo de tarefa com opções avançadas.

Depois de reunir os dados de todas as estações de trabalho da empresa, eles copiam os dados resultantes em uma única estação de trabalho e os processam para gerar um relatório. Eles escolhem dinamicamente a melhor estação de trabalho disponível definindo os requisitos necessários para executar a tarefa: uma estação de trabalho com bastante espaço em disco, CPU poderosa e o programa necessário para gerar o relatório.

Se o administrador não quiser modificar o fluxo de tarefas que usou antes do Tivoli Workload Scheduler versão 8.6 para executar a tarefa Java, por exemplo, é possível modificar o nome da estação de trabalho onde ele deseja executar a tarefa, inserindo o nome de um conjunto ou conjunto dinâmico do agentes dinâmicos onde o arquivo executável Java está instalado. O Tivoli Workload Scheduler converte a sintaxe da tarefa, para que ela possa ser executada pelo programa Java e designa a tarefa para o melhor recurso disponível no conjunto.

O relatório destaca quantos contratos novos foram assinados e quantos clientes estão com os pagamentos atrasados. Uma correspondência é enviada ao contador chefe, listando o número de novos contratos e clientes em atraso.

A empresa pode atingir esse objetivo desta maneira:

- Usando as novas estações de trabalho com capacidades dinâmicas para executar as tarefas que o administrador criou para as estações de trabalho existentes do Tivoli Workload Scheduler. Para executar essas tarefas nas novas estações de trabalho, o administrador altera apenas a estação de trabalho onde deseja executar a tarefa. A principal vantagem é que ele pode usar os fluxos de tarefas que criou anteriormente sem esforço adicional.
- Definindo diversos tipos de tarefas com opções avançadas sem precisar especificar qualificações específicas sobre os aplicativos nos quais a tarefa é executada.

Esses tipos de tarefas com opções avançadas são executados nas seguintes estações de trabalho:

agentes dinâmicos
: Estações de trabalho capazes de executar as tarefas existentes e os tipos de tarefas com opções avançadas.

Conjuntos
: Grupos nos quais é possível incluir agentes dinâmicos dependendo de suas necessidades. As tarefas são designadas dinamicamente para o melhor agente disponível.

Conjuntos dinâmicos
: Grupos de agentes dinâmicos para os quais você especifica seus requisitos e deixa o Tivoli Workload Scheduler selecionar os agentes dinâmicos que atendem às suas necessidades. As tarefas são designadas dinamicamente para o melhor agente dinâmico disponível.

Cenário: Criando uma Definição de Tarefa e Enviando para um Conjunto Dinâmico

Nesse cenário, você define os requisitos para executar a tarefa ao criar o conjunto dinâmico, por exemplo, é possível incluir no conjunto dinâmico todas as estações de trabalho com sistema operacional Windows e DB2 instalados e utilização máxima de CPU 50%. O conjunto dinâmico é então preenchido com as estações de trabalho que correspondem aos seus requisitos e está pronto para o envio da tarefa.

Para criar um conjunto dinâmico e enviar uma tarefa para ele, execute as seguintes tarefas:

1. Efetue login no Dynamic Workload Console.
2. Na barra de ferramentas de navegação, clique em **Administração > Design do Ambiente de Carga de Trabalho > Criar Estações de Trabalho**.
3. Selecione um mecanismo e clique em **OK**.
4. A página **Propriedades da Estação de Trabalho** é exibida.
5. Selecione **Conjunto Dinâmico** no menu **Tipo de estação de trabalho**.
6. Preencha os campos requeridos.
7. Clique em **Editar Requisitos**. A página **Requisitos** é exibida.
8. Especifique os seguintes requisitos:
   - Selecione o sistema operacional na área de janela **Sistema Operacional**.
   - Selecione a utilização de CPU na área de janela **Utilização de CPU**.
   - Selecione o recurso lógico requerido na área de janela **Recursos lógicos**. Para incluir estações de trabalho com o DB2 instalado, clique em **Incluir** e especifique o seu requisito na área de janela **Requisitos**.
   - Opcionalmente, selecione a política de otimização requerida.
9. Clique em **OK** para salvar os seus requisitos.
10. Clique em **Salvar** para salvar o conjunto dinâmico.
11. Na barra de ferramentas de navegação, clique em **Administração** > **Design de Carga de Trabalho** > **Gerenciar Definições de Carga de Trabalho**
12. Especifique um nome do mecanismo, distribuído ou z/OS. O Workload Designer é aberto. Os tipos de tarefas e as características variam, dependendo de se você selecionar um mecanismo distribuído ou um z/OS.
13. Selecione **Novo > Definição de Tarefa > Banco de Dados e Integrações > Banco de Dados**.
14. Preencha os campos obrigatórios e especifique o conjunto dinâmico que você criou previamente no campo **Estação de Trabalho**.
15. Digite as suas instruções SQL na guia **SQL**.
16. Clique em **Salvar** para salvar a tarefa.
17. Na barra de ferramentas de navegação, clique em **Administração** > **Envio de Carga de Trabalho** > **Enviar Tarefas Predefinidas**.

Cenário: Criando uma Definição de Tarefa e Enviando para um Conjunto

Nesse cenário, você precisa executar um script de atualização de inventário, portanto, você cria um conjunto, agrupando estações de trabalho onde o programa requerido está instalado, e uma tarefa para executar o script. Você seleciona o sistema de destino para a tarefa do conjunto`invadmin`.

Para criar um conjunto e enviar uma tarefa para ele, execute as seguintes etapas:

1. Efetue login no Dynamic Workload Console.
2. Na barra de ferramentas de navegação, clique em **Administração > Design do Ambiente de Carga de Trabalho > Criar Estações de Trabalho**.
3. Selecione um mecanismo e clique em **OK**.
4. A página **Propriedades da Estação de Trabalho** é exibida.
5. Selecione **Conjunto** no menu **Tipo de estação de trabalho**.
6. Nomeie o conjunto `invadmin` e preencha os campos necessários.
7. Selecione as estações de trabalho que deseja incluir no conjunto. Selecione as estações de trabalho nas quais o programa requerido está instalado.
8. Clique em **Salvar** para salvar o conjunto.
9. Na barra de ferramentas de navegação, clique em **Administração** > **Design de Carga de Trabalho** > **Gerenciar Definições de Carga de Trabalho**
10. Especifique um nome do mecanismo, distribuído ou z/OS. O Workload Designer é aberto. Os tipos de tarefas e as características variam, dependendo de se você selecionar um mecanismo distribuído ou um z/OS.
11. Selecione **Novo > Definição de Tarefa > Nativo > Executável**.
12. Especifique um nome para a tarefa.
13. No campo **Estação de trabalho**, especifique o conjunto`invadmin` que você definiu previamente.
14. Navegue para a guia **Tarefa** e selecione **Comando**.
15. Digite o nome e o caminho para o arquivo executável que deseja executar, localizado em cada estação de trabalho no conjunto.
16. Clique em **Salvar** para salvar a tarefa.
17. Na barra de ferramentas de navegação, clique em **Administração** > **Envio de Carga de Trabalho** > **Enviar Tarefas Predefinidas**.

Limitações para tarefas no fluxo de tarefas `USERJOBS` no planejamento dinâmico

Recursos e propriedades parcialmente suportados(as) ou não suportados(as) para o fluxo de tarefas*USERJOBS* no planejamento dinâmico

As tarefas do fluxo de tarefas *USERJOBS* suportam a maioria dos recursos ou propriedades de planejamento dinâmico. A Tabela 77 lista alguns recursos ou propriedades que são parcialmente suportados(as) ou não suportados(as).

Tabela 77. Recursos parcialmente ou não suportados para tarefas no fluxo de tarefas USERJOBS

Recurso\Propriedade	*USERJOBS*
Os tipos de tarefas com opções avançadas podem perder algumas informações ao serem movidos para o fluxo de tarefas *USERJOBS*.	O tipo de tarefa pode estar ausente ou diferente do original.
Exibição dos tipos de tarefas com opções avançadas de propriedades usando a linha de comandos **conman**.	Se você executar **sj <job_name>; props**, em que <*job_name*> é o tipo de tarefa com o nome das opções avançadas, você terá os seguintes problemas: • O valor do campo `Tarefa` da seção `Informações Gerais` fica truncado. **Nota:** Esse problema afeta apenas a exibição da tarefa e não afeta a própria tarefa. • A seção `Informações Extras` exibe o nome interno das propriedades, em vez de exibir o nome externo traduzido.
Exibe os tipos de tarefas com opções avançadas usando o Dynamic Workload Console.	Se você utilizar as visualizações gráficas do Dynamic Workload Console, tarefas sombra não serão exibidas com os pontos típicos.
Variáveis transmitidas entre tarefas na mesma instância do fluxo de tarefas. Para obter mais informações sobre esse recurso, consulte "Passando Variáveis Entre Tarefas na Mesma Instância do Fluxo de Tarefas" na página 480.	Esse recurso não é suportado. A variável da tarefa transmitida não é resolvida quando as tarefas são movidas do fluxo de tarefas original para o fluxo de tarefas *USERJOBS*.

Capítulo 13. Utilizando Comandos Utilitários

Este capítulo descreve os comandos utilitários do Tivoli Workload Scheduler. Esses comandos, com as exceções listadas abaixo, são instalados no diretório *TWS_home*/bin. Os comandos utilitários são executados a partir do prompt de comandos do sistema operacional.

O **StartUp** é instalado no diretório *TWS_home* e **version** é instalado no diretório *TWS_home*/version.

Descrições dos comandos

A Tabela 78 contém a lista dos comandos utilitários e, para cada um, sua descrição e os sistemas operacionais suportados.

Tabela 78. Lista de Comandos Utilitários

Comando	Descrição	Sistema Operacional
at	Envia uma tarefa para ser executada em um período específico.	UNIX
batch	Envia uma tarefa para ser executada o mais breve possível.	UNIX
cpuinfo	Retorna informações de uma definição de estação de trabalho.	UNIX, Windows
datecalc	Converte data e hora em um formato necessário.	UNIX, Windows
delete	Remove os arquivos de script e arquivos de lista padrão por nome.	UNIX, Windows
evtdef	Importa/exporta definições de evento customizado.	UNIX, Windows
evtsize	Define o tamanho máximo dos arquivos de mensagens de eventos.	UNIX, Windows
exportservedata	Efetua o download da lista de instâncias do broker de carga de trabalho dinâmica a partir do banco de dados Tivoli Workload Scheduler e muda um número da porta ou um nome do host.	UNIX, Windows
importservedata	Efetua o upload da lista de instâncias do broker de carga de trabalho dinâmica para o banco de dados Tivoli Workload Scheduler após editar o arquivo temporário para mudar um número da porta ou um nome do host.	UNIX, Windows
jobinfo	Retorna informações sobre uma tarefa.	UNIX, Windows
jobstdl	Retorna os nomes de caminhos de arquivos de lista padrão.	UNIX, Windows
listproc	Lista processos. Esse comando não é suportado.	Windows
killproc	Elimina processos. Esse comando não é suportado.	Windows
maestro	Retorna o diretório home do Tivoli Workload Scheduler.	UNIX, Windows
makecal	Cria calendários personalizados.	UNIX, Windows
metronome.pl	É substituído por **tws_inst_pull_info**.	UNIX, Windows
morestdl	Exibe o conteúdo de arquivos de lista padrão.	UNIX, Windows
movehistorydata	Move os dados presentes no banco de dados Tivoli Workload Scheduler para as tabelas de archive.	UNIX, Windows
param	Cria, exibe e exclui variáveis e senhas do usuário em agentes dinâmicos.	UNIX, Windows
parms	Exibe, altera e adiciona parâmetros.	UNIX, Windows
release	Libera unidades de um recurso.	UNIX, Windows
rmstdlist	Remove arquivos de lista padrão com base no tempo.	UNIX, Windows
sendevent	Envia os eventos genéricos ao servidor do processador de eventos atualmente ativo.	UNIX, Windows
showexec	Exibe informações sobre as tarefas em execução.	UNIX
shutdown	Para o processo **netman** e, opcionalmente, WebSphere Application Server.	UNIX, Windows
ShutDownLwa	Para o agent localmente.	UNIX, Windows **Nota:** Em sistemas UNIX, ele pode ser executado apenas por *TWS_user* ou pelo usuário root.
StartUp	Inicia o processo **netman** e, opcionalmente, WebSphere Application Server.	UNIX, Windows

© Copyright IBM Corp. 1999, 2014

Tabela 78. Lista de Comandos Utilitários (continuação)

Comando	Descrição	Sistema Operacional
StartUpLwa	Inicia o agent localmente.	UNIX, Windows **Nota:** Em sistemas UNIX, ele pode ser executado apenas por *TWS_user* ou pelo usuário root.
tws_inst_pull_info	Coleta dados na instância e estação de trabalho local do Tivoli Workload Scheduler, no WebSphere Application Server e no DB2 para propósitos de diagnóstico. É documentado em *Tivoli Workload Scheduler: Troubleshooting Guide*.	UNIX, Windows
versão	Exibe informações sobre a versão.	UNIX

at e batch

Enviam comandos e tarefas ad hoc para serem ativados pelo Tivoli Workload Scheduler.

Esses comandos são executados somente no UNIX.

Consulte **at.allow** e **at.deny** a seguir para obter informações sobre a disponibilidade aos usuários.

Sintaxe

at -V | -U

at {-s *jstream* **| -q** *queue***}** *time-spec*

batch -V | -U

batch [**-s** *jstream*]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

-s *jstream*

Especifica o *jobstream_id* da instância do fluxo de tarefas na qual a tarefa é enviada. Se uma instância de fluxo de tarefas com esse *jobstream_id* não existir, é criada uma nova instância de tarefa tendo *jstream* como alias e como *jobstream_id*. O nome deve iniciar com uma letra e pode conter caracteres alfanuméricos e traços. Ele pode conter até 16 caracteres.

Se os argumentos **-s** e **-q** forem omitidos, um nome de fluxo de tarefas será selecionado com base no valor da variável de ambiente ***ATSCRIPT***. Se ***ATSCRIPT*** contiver a palavra **maestro**, o alias do fluxo de tarefas será os oito primeiros caracteres do nome do grupo do usuário. Se ***ATSCRIPT*** não for configurado ou for configurado com um valor diferente de **maestro**, o alias do fluxo de tarefas será **at** (para tarefas enviadas com **at**) ou **batch** (para tarefas enviadas com **batch**).

Consulte “Outras Considerações” na página 496 para obter informações adicionais sobre fluxos de tarefas.

As palavras-chave a seguir aplicam-se somente às tarefas **at**:

-q*queue*

Especifica para enviar a tarefa a um fluxo de tarefas com o nome *queue*, que pode ser uma única letra (a-z). Consulte "Outras Considerações" na página 496 para obter informações adicionais sobre fluxos de tarefas.

time-spec

Especifica a hora em que a tarefa será ativada. A sintaxe é a mesma utilizada com o comando UNIX **at**.

Comentários

Após digitar **at** ou **batch**, digite os comandos que constituem a tarefa. Termine cada linha de entrada pressionando a tecla Return. A seqüência inteira termina com fim-de-linha (geralmente, **Control+d**), ou digitando uma linha com um ponto (.). Alternativamente, utilize um sinal de menor (<) para ler os comandos de um arquivo. Consulte o "Exemplos".

São enviadas informações sobre as tarefas **at** e **batch** ao gerenciador de domínio principal no qual as tarefas são incluídas nos fluxos de tarefas no plano de produção, arquivo `Symphony`. As tarefas são ativadas com base nas dependências incluídas nos fluxos de tarefas.

O shell UNIX utilizado para tarefas enviadas com os comandos **at** e **batch** é determinado pela variável ***SHELL_TYPE*** no script de configuração `jobmanrc`. Não utilize o shell C. Para obter informações adicionais, consulte a "Customizando o Processamento da Tarefa em uma Estação de Trabalho UNIX - jobmanrc" na página 48.

Depois de enviadas, as tarefas são ativadas da mesma maneira que as outras tarefas planejadas. Cada tarefa é executada no ambiente do usuário que está enviando. Para assegurar que o ambiente esteja completo, são inseridos comandos **set** no script para corresponder às definições de variáveis no ambiente do usuário.

Exemplos

Para enviar uma tarefa em um fluxo de tarefas com *jobstream_id* `sched8` para ser ativada assim que possível, execute o seguinte comando:

```
batch -s sched8
command <Return>
...
<Control d>
```

Para enviar uma tarefa para ser ativada duas horas depois da hora em que o comando foi digitado, execute o seguinte comando:

```
at now + 2 hours
command <Return>
...
<Control d>
```

Se a variável ***ATSCRIPT*** for nula, a tarefa é enviada em um fluxo de tarefas com o mesmo nome que o grupo do usuário. Caso contrário, ele será submetido a um fluxo de tarefa chamado `at`.

Para enviar uma tarefa em uma instância de fluxo de tarefas com *jobstream_id* `sked-mis` para ser ativada às 17h30, execute o seguinte comando:

```
at -s sked-mis 17h30
command <Return>
...
<Control d>
```

O comando a seguir é o mesmo do comando anterior, exceto que os comandos da tarefa são lidos em um arquivo:

```
at -s sked-mis 17h30 < ./myjob
```

O fato dos comandos serem lidos de um arquivo não altera seu modo de processamento. Isto é, os comandos são copiados do arquivo `./myjob` para um arquivo de script.

Substituindo os Comandos do UNIX

Os comandos padrão do UNIX, **at** e **batch**, podem ser substituídos por comandos do Tivoli Workload Scheduler. Os comandos a seguir mostram como substituir os comandos **at** e **batch** do UNIX:

```
$ mv /usr/bin/at /usr/bin/uat
$ mv /usr/bin/batch /usr/bin/ubatch
$ ln -s TWShome/bin/at /usr/bin/at
$ ln -s TWShome/bin/batch /usr/bin/batch
```

Os Arquivos at.allow e at.deny

Os comandos **at** e **batch** utilizam os arquivos `/usr/lib/cron/`**`at.allow`** e `/usr/lib/cron/`**`at.deny`** para restringir o uso. Se o arquivo `at.allow` existir, somente os usuários listados no arquivo terão permissão para utilizar **at** e **batch**. Se o arquivo não existir, `at.deny` será verificado para saber se o usuário teve a permissão negada explicitamente. Se nenhum dos arquivos existir, apenas o usuário **root** terá permissão para utilizar os comandos.

Arquivos de Script

Os comandos digitados com **at** ou **batch** são armazenados em arquivos de script. Os arquivos são criados pelo Tivoli Workload Scheduler utilizando a seguinte convenção de nomenclatura:

TWS_home`/atjobs/`***epoch.sss***

em que:

epoch O número de segundos desde 00:00, 1/1/70.

sss Os três primeiros caracteres do nome do fluxo de tarefas.

Nota: O Tivoli Workload Scheduler remove os arquivos de script para tarefas que não têm o saldo transportado. No entanto, é recomendável monitorar o espaço em disco no diretório `atjobs` e remover os arquivos antigos, se necessário.

Nomes de Tarefas

Todas as tarefas **at** e **batch** recebem nomes exclusivos do Tivoli Workload Scheduler quando são enviadas. Os nomes consistem no ID do processo (PID) do usuário precedido pelo nome do usuário truncado de forma que não exceda oito caracteres. O nome resultante é organizado em ordem crescente.

Outras Considerações

- Os fluxos de tarefas aos quais as tarefas **at** e **batch** são enviadas devem ser criados antecipadamente com o composer. Os fluxos de tarefas podem conter dependências que determinam quando as tarefas serão ativadas. No mínimo, os fluxos de job devem conter a palavra-chave **carryforward**. Isso assegura que as tarefas que não são concluídas ou não são ativadas enquanto o plano de produção atual está em processo tenham o saldo transportado para o próximo plano de produção.
- Inclua a expressão **on everyday** para que os fluxos de tarefas sejam selecionados todos os dias.

- Utilize a palavra-chave **limit** para limitar o número de tarefas enviadas que podem ser executadas simultaneamente.
- Utilize a palavra-chave **priority** para definir a prioridade de tarefas enviadas em relação a outras tarefas.

Se o valor de tempo for menor do que a hora atual, ele será considerado como para o dia seguinte. Se o valor de tempo for maior do que a hora atual, ele será considerado como para o dia atual.

cpuinfo

Retorna informações de uma definição de estação de trabalho.

Sintaxe

cpuinfo -V | -U

cpuinfo *workstation* [*infotype*] [...]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

estação de trabalho

O nome da estação de trabalho.

infotype

O tipo de informação a ser exibida. Especifica um ou mais dos seguintes:

os_type

Retorna o valor do campo **os**: **UNIX**, **WNT**, **ZOS**, **OTHER** e **IBM i**. O valor **ZOS** aplica-se somente a estações de trabalho de mecanismo remoto usadas para se comunicarem com um controlador do Tivoli Workload Scheduler for z/OS.

nó Retorna o valor do campo **node**. Para um servidor do Workload Broker, é o nome do host ou o endereço TCP/IP da estação de trabalho onde o Tivoli Workload Scheduler Bridge foi instalado. Para uma estação de trabalho do mecanismo remoto, é o nome do host da estação de trabalho onde o mecanismo remoto está instalado. Em qualquer outro caso, especifique o nome do host ou o endereço TCP/IP da estação de trabalho.

porta Retorna o valor do campo **tcpaddr**. Se você estiver definindo uma estação de trabalho do workload broker, especifique o valor da propriedade **TWS.Agent.Port** do arquivo TWSAgentConfig.properties. Para estações de trabalho de mecanismo remoto, o valor desse campo ;e o número da porta HTTP usado pelo mecanismo remoto. Se o protocolo HTTPS for usado, o valor desse campo é 31111.

sslport

Retorna o valor do campo **secureaddr**. É a porta usada para atender conexões SSL recebidas. Para estações de trabalho de mecanismo remoto, o valor desse campo ;e o número da porta HTTPS usado pelo mecanismo remoto. Se o protocolo HTTP for usado, o valor desse campo é 31113.

engineaddr
Para qualquer tipo de estações de trabalho, o valor desse campo é 0.

protocol
Retorna o valor do campo **protocol**: HTTP ou HTTPS. Quando o tipo de estação de trabalho é mecanismo remoto, esse valor indica o protocolo usado para se comunicar entre o servidor broker e o mecanismo remoto.

sec_level
Retorna o valor do campo **securitylevel**: NONE, ENABLED, ON ou FORCE.

autolink
Retorna o valor do campo **autolink**: ON ou OFF.

fullstatus
Retorna o valor do campo **fullstatus**: ON ou OFF.

resolvedep
Retorma ON ou OFF. Não é mais usado na versão 8.6.

behindfirewall
Retorna o valor do campo **behindfirewall**: ON ou OFF.

host Retorna o valor do campo **host**. É o nome da estação de trabalho hosting o agente.

domínio
Retorna o valor do campo **domain**.

ID Retorna o identificador do agente usado pela estação de trabalho ao conectar ao servidor broker. Para estação de trabalho com tipo: AGENT, REM-ENG, POOL, D-POOL.

método
Somente para agentes estendidos e de rede. Retorna o valor do campo **access**.

servidor
Retorna o valor do campo **server**.

tipo Retorna o valor do campo **type**. Mostra o tipo de estação de trabalho: MASTER, MANAGER, FTA, S-AGENT, REM-ENG, AGENT, POOL, D-POOL e X-AGENT.

time_zone
Retorna o valor do ccampo **timezone**. Mostra o fuso horário da estação de trabalho. Para um agente estendido, o campo fica em branco. Para uma estação de trabalho de mecanismo remoto, esse é o fuso horário do mecanimso remoto.

version
Retorna a versão do Tivoli Workload Scheduler que está em execução na estação de trabalho. Para um agente estendido, o campo fica em branco.

info Retorna a versão do sistema operacional e o modelo de estação de trabalho. Para agentes estendidos, o campo está em branco. Para estações de trabalho do mecanismo remoto, esse campo exibe Mecanismo Remoto.

Comentários

Os valores são retornados, um em cada linha, na mesma ordem em que os argumentos foram digitados na linha de comandos. Se nenhum argumento for especificado, todas as informações aplicáveis serão retornadas com etiquetas, uma em cada linha.

Exemplos

Os exemplos a seguir baseiam-se na seguinte definição de estação de trabalho:

```
Workstation Name Type  Domain   Updated On Locked By
---------------  ------- ---------------  ----------  ------------------
RE-ZOS   REM-ENG -     09/06/2010 -

CPUNAME RE-ZOS
  OS ZOS
  NODE 9.168.119.189 TCPADDR 635
  FOR MAESTRO HOST NC123162_DWB
    TYPE REM-ENG
    PROTOCOL HTTP
END
```

Para imprimir **tipo** e **protocolo** para a estação de trabalho RE-ZOS, execute o comando a seguir:

```
>cpuinfo RE-ZOS type protocol
REM-ENG
HTTP
```

Para imprimir todas as informações para a estação de trabalho RE-ZOS, execute o comando a seguir:

```
>cpuinfo RE-ZOS
 OS_TYPE: ZOS
 NODE: 9.168.119.189
 PORT: 635
 SSLPORT: 31113
 ENGINEADDR: 0
 PROTOCOL: HTTP
 AUTOLINK: OFF
 FULLSTATUS: OFF
 RESOLVEDEP: OFF
 BEHINDFIREWALL: OFF
 HOST: NC123162_DWB
 DOMAIN: MASTERDM
 ID: D795263CBCD2365CA7B5C5BC0C3DD363
 SERVER:
 TYPE: REM-ENG
 TIME_ZONE: Europe/Rome
 VERSION: 8.6
 INFO: Remote Engine
```

datecalc

Resolve expressões de datas e retorna datas nos formatos escolhidos.

Sintaxe

datecalc -V | -U

datecalc *base-date*
[*offset*]
[**pic** *format*]
[**freedays** *Calendar_Name* [**-sa**] [**-su**]]

datecalc -t *time*
[*base-date*]
[*offset*]
[**pic** *format*]

datecalc *yyyymmddhhtt*
[*offset*]
[**pic** *format*]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

base-date

Especifique um dos seguintes:

day | *date* | **today** | **tomorrow** | **scheddate**

em que:

day Especifica um dia da semana. Os valores válidos são: **su**, **mo**, **tu**, **we**, **th**, **fr** ou **sa**.

date Especifica uma data, no formato *element*/*element*[/*element*], em que *element* é: *d*[*d*], *m*[*m*] e *yy*[*yy*]. Qualquer formato diferente de data não é válido.

Se forem utilizados dois dígitos para o ano (*yy*), um número superior a 70 será uma data do século 20 e um número inferior a 70 será uma data do século 21.

O parâmetro se refere à data real, não ao comando *date* do UNIX. O exemplo a seguir mostra uma opção para usar a saída da *data* do UNIX como entrada para o parâmetro *date* do Tivoli Workload Scheduler.

```
hdate=´date +"%m/%d/%y"´
echo $hdate
datecalc $hdate pic mm/dd/yyyy
```

Os valores válidos para o mês (*m*[*m*]) são **jan**, **feb**, **mar**, **apr**, **may**, **jun**, **jul**, **aug**, **sep**, **oct**, **nov** ou **dec**.

As barras (/) podem ser substituídas por traços (-), pontos (.), vírgulas (,) ou espaços. Por exemplo, qualquer um dos seguintes pode ser digitado para 28 de março de 2005:

03/28/05
3-28-2005
28.mar.05
05,28,3
mar 28 2005
28 3 05

Se forem utilizados números, é possível que a data digitada seja ambígua, por exemplo, 2,7,04. Neste caso, **datecalc** utiliza o formato de data definido no catálogo de mensagens do Tivoli Worload Scheduler para interpretá-la. Se a data não corresponder ao formato, uma mensagem de erro será gerada por **datecalc**.

today Especifica a data atual do sistema.

tomorrow
Especifica a data atual do sistema mais um dia, ou, no caso de cálculos de horas, mais 24 horas.

scheddate
Especifica a data do plano de produção. Pode não ser a mesma data do sistema. Quando utilizado dentro de tarefas em um fluxo de tarefas que não é um fluxo de tarefas transportado, retorna a

data de quando a tarefa deve ser executada, que poderia ser diferente da data de produção do fluxo de tarefas se a tarefa tiver uma dependência `at` especificada.

Quando utilizado dentro de tarefas em um fluxo de tarefas transportado, retorna a data de quando a tarefa deveria ter sido executada, que poderia ser diferente da data de produção do fluxo de tarefas transportado se a tarefa tiver uma dependência `at` especificada. Se a dependência `at` for utilizada com a seguinte sintaxe: **at**=*hhmm* + *n* **days**, *n* **days** não serão incluídos na variável ***TIVOLI_JOB_DATE*** e, portanto, o comando **datecalc** não relatará esses dias.

-t *time* **[***base-date***]**
Especifique *time* em um dos seguintes formatos:

now | **noon** | **midnight** | [*h*[*h*][[:]*mm*] [**am** | **pm**] [**zulu**]

em que:

now Especifica a data e hora atuais do sistema.

noon Especifica 12h (ou 1200).

midnight
Especifica 24h (ou 2400).

h[*h*]**[[:]***mm***]**
Especifica a hora e os minutos em 12 horas (se **am** ou **pm** for utilizado), ou 24 horas. O delimitador opcional de dois-pontos (:) pode ser substituído por um ponto (.), uma vírgula (,), um apóstrofo ('), pela letra **h**, ou por um espaço. Por exemplo, qualquer um dos seguintes pode ser digitado para 20h:

8:00pm
20:00
0800pm
2000
20h
20
8,00pm
20.00
8\'00pm
20 00

zulu Especifica que a hora digitada é a hora de Greenwich (Hora Coordenada Universal). **datecalc** a converte na hora local.

yyyymmddhhtt
Especifica o ano, mês, dia, hora e minuto expresso em exatamente doze dígitos. Por exemplo, para 2005, 7 de maio, 9h15, digite o seguinte: **200505070915**

deslocamento
Especifica um deslocamento de *base-date* no seguinte formato:

{**[+** | **>** | **-** | **<** *number* | **nearest]** | **next**} **day[s]** | **weekday[s]** | **workday[s]** | **week[s]** | **month[s]** | **year[s]** | **hour[s]** | **minute[s]** | *day* | *calendar*

em que:

+ | > Especifica um deslocamento para uma data ou hora posterior. Utilize + (Mais) no Windows; utilize > (maior que) no UNIX. Certifique-se de incluir uma barra invertida (\) antes do sinal de maior (>).

- | < Especifica um deslocamento para uma data ou hora anterior. Utilize - (Menos) no Windows; utilize < (menor que) no UNIX. Certifique-se de incluir uma barra invertida (\) antes do sinal de maior (>).

número
O número de unidades do tipo especificado.

nearest
Especifica um deslocamento para a ocorrência mais próxima do tipo de unidade (anterior ou posterior).

next Especifica a próxima ocorrência do tipo de unidade.

day[s] Especifica todos os dias.

weekday[s]
Especifica todos os dias, exceto sábado e domingo.

workday[s]
O mesmo de **weekday[s]**, mas também exclui as datas do calendário**holidays**.

week[s]
Especifica sete dias.

month[s]
Especifica os meses do calendário.

year[s]
Especifica os anos do calendário.

hour[s]
Especifica as horas do relógio.

minute[s]
Especifica os minutos do relógio.

day Especifica um dia da semana. Os valores válidos são: **su**, **mo**, **tu**, **we**, **th**, **fr** ou **sa**.

calendário
Especifica as entradas em um calendário com este nome.

pic *format*
Especifica o formato em que a data e hora são retornados. Os caracteres de *format* são os seguintes:

m Número do mês.

d Número do dia.

y Número do ano.

j Número do dia Juliano.

h Número da hora.

t Número do minuto.

^|/ Um espaço. Utilize / (barra) no Windows; utilize ^ (acento

circunflexo) no UNIX (inclua uma barra invertida (\) antes do acento circunflexo (^) se estiver no shell Bourne).

Caracteres de pontuação também podem ser incluídos. Eles funcionam da mesma forma que os delimitadores utilizados em *date* e *time*.

Se um formato não for definido, **datecalc** retornará a data e hora no formato definido pelas variáveis de ambiente NLS (Native Language Support). Se as variáveis NLS não forem definidas, o idioma nativo será padronizado como C.

dias livres

Especifica o nome de um calendário de dias livres *Nome_do_Calendário* que deve substituir **feriados** na avaliação de *dias úteis*.

Nesse caso, *dias úteis* é avaliado como *todos os dias* excluindo *sábado*, *domingo* e todas as datas listadas em *Nome_do_Calendário*.

Por padrão, *sábado* e *domingo* não são considerados *dias úteis*, a não ser que seja explicitamente especificado de outra forma incluindo-se **-sa** e **-su** depois de *Calendar_Name*.

Também é possível especificar **feriados** como o nome do calendário de dias livres.

Exemplos

Para retornar a próxima data, a partir de hoje, no calendário `monthend`, execute o seguinte comando:

```
>datecalc today next monthend
```

Nos exemplos a seguir, a data atual do sistema é sexta-feira, 16 de abril de 2006.

```
>datecalc today +2 days pic mm/dd/yyyy
04/16/2006

>datecalc today next tu pic yyyy\^mm\^dd
2006 04 16

>LANG=american;export LANG
>datecalc -t 14:30 tomorrow
Sat, Apr 17, 2006 02:30:00 PM

>LANG=french;datecalc -t 14:30 tomorrow
Samedi 17 avril 2006 14:30:00
```

No exemplo a seguir, a hora atual do sistema é 10:24h.

```
>datecalc -t now \> 4 hours pic hh:tt
14:24
```

datamigrate

Use o utilitário **datamigrate** no gerenciador de domínio principal para importar para o banco de dados os dados que são salvos nos arquivos simples criados utilizando a linha de comandos **composer** em outra instância do Tivoli Workload Scheduler.

Sintaxe

Use a seguinte sintaxe e ordem para importar dados:

datamigrate -u | -v

datamigrate -<*object_type*> <*object_type_file*> [-tmppath <temporary_directory]

Argumentos

-u Exibe informações de utilização do comando e sai.

-v Exibe a versão do comando e sai.

-<*object_type*> **<object_type_file>**

-topology *topology_filename*
Importa todas as definições de domínio, de estação de trabalho e de classe de estação de trabalho salvas no arquivo *topology_filename* que é criado utilizando a linha de comandos **composer** na instância do Tivoli Workload Scheduler.

-prompts *prompts_filename*
Importa todas as definições de prompt salvas no arquivo *prompts_filename* que é criado utilizando a linha de comandos **composer** na instância do Tivoli Workload Scheduler.

-calendars *calendars_filename*
Importa todas as definições de calendário salvas no arquivo *calendars_filename* que é criado utilizando a linha de comandos **composer** na instância do Tivoli Workload Scheduler.

-parms *parms_filename*
Importa todas as definições de parameter salvas no arquivo *parms_filename* que é criado utilizando a linha de comandos **composer** na instância do Tivoli Workload Scheduler.

-resources *resources_filename*
Importa todas as definições de recurso salvas no arquivo *resources_filename* que é criado usando a linha de comandos **composer** na instância do Tivoli Workload Scheduler.

-rcgroups *rcgroups_filename*
Importa todas as definições de grupo do ciclo de execução salvas no arquivo *rcgroups_filename* que é criado usando a linha de comandos **composer** na instância do Tivoli Workload Scheduler.

-users *users_filename*
Importa todas as definições de usuário salvas no arquivo *users_filename* que é criado usando a linha de comandos **composer** na instância do Tivoli Workload Scheduler.

Nota: Certifique-se de substituir no arquivo de usuários, todas as entradas `"*********"` pelos valores de senha reais antes de executar o utilitário **datamigrate**.

-jobs *jobs_filename*
Importa todas as definições de tarefa salvas no arquivo *jobs_filename* que é criado usando a linha de comandos **composer** na instância do Tivoli Workload Scheduler.

-scheds *scheds_filename*
Importa todas as definições de fluxo de tarefas (planejamento) salvas no arquivo *scheds_filename* que é criado usando a linha de comandos **composer** na instância do Tivoli Workload Scheduler.

-tmppath *temp_path*
temp_path é o caminho temporário onde **datamigrate** pode colocar seus

arquivos de trabalho temporários. O padrão é <TWS_installation_dir>\tmp em sistemas operacionais Windows e <TWS_installation_dir>/tmp em sistemas UNIX.

Comentários

Se preferir usar o utilitário **datamigrate** em vez do comando **composer** com as opções add ou replace, certifique-se de seguir a ordem correta para importar os dados.

Exemplos

Para importar para o banco de dados do Tivoli Workload Scheduler V9.2 os dados exportados a partir do Tivoli Workload Scheduler V8.6, execute na seguinte ordem:

```
datamigrate -topology topology86.txt
datamigrate -prompts prompt86.txt
datamigrate -calendars cals86.txt
datamigrate -parms parms86.txt
datamigrate -resources res86.txt
datamigrate -rcgroups rcs86.txt
datamigrate -users users86.txt
datamigrate -jobs jobs86.txt
datamigrate -scheds js86.txt
```

Excluir

Remove arquivos. Embora o objetivo desse arquivo seja remover arquivos de lista padrão, sugerimos a utilização do comando rmstdlist no lugar. Os usuários **maestro** e **root** no UNIX e **Administrator** no Windows podem remover qualquer arquivo. Outros usuários podem remover somente os arquivos associados a seus próprios jobs.

Sintaxe

delete -V | -U

delete *filename*

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

nome do arquivo

Especifica o nome do arquivo ou grupo de arquivos a serem removidos. O nome deverá ser colocado entre aspas (") se contiver caracteres diferentes do seguinte: alfanuméricos, traços (-), barras (/), barras invertidas (\) e sublinhados (_). Caracteres curinga são permitidos.

Nota: Utilize esse comando com cuidado. O uso inadequado de caracteres curinga poderá resultar na remoção acidental de arquivos.

Exemplos

Para remover todos os arquivos de lista padrão em 4/11/04:

```
delete d:\win32app\maestro\stdlist\2004.4.11\@
```

O script a seguir, incluído em uma tarefa planejada no UNIX, remove o arquivo de lista padrão da tarefa se não houver nenhum erro:

```
...
#Remover a stdlist deste job:
if grep -i error $UNISON_STDLIST
then
exit 1
else
`maestro`/bin/delete $UNISON_STDLIST
fi
...
```

O script de configuração padrão, `jobmanrc`, define a variável ***UNISON_STDLIST*** com o nome do arquivo de lista padrão do job. Para obter informações adicionais sobre `jobmanrc`, consulte "Customizando o Processamento da Tarefa em uma Estação de Trabalho UNIX - jobmanrc" na página 48.

evtdef

Importa/exporta um arquivo de definição XML do provedor de eventos genéricos em que é possível incluir e modificar tipos de evento customizado. É possível, então, utilizar o comando **sendevent** para enviar esses eventos para o servidor de processamento de eventos. Consulte também "Definindo Eventos Customizados" na página 134.

Sintaxe

evtdef -U | -V

evtdef [*connection parameters*] **dumpdef** *file-path*

evtdef [*connection parameters*] **loaddef** *file-path*

Argumentos

-U Exibe informações de utilização do comando e sai.

-V Exibe a versão do comando e sai.

parâmetros de conexão

Se estiver usando **evtdef** a partir do gerenciador de domínio principal, os parâmetros de conexão foram configurados na instalação e não precisam ser fornecidos, a menos que você não deseje usar os valores padrão.

Se estiver usando **evtdef** a partir do cliente da linha de comandos em outra estação de trabalho, os parâmetros de conexão podem ser fornecidos por um ou mais destes métodos:

- Armazenados no arquivo `localopts`
- Armazenados no arquivo `useropts`
- Fornecidos ao comando em um arquivo de parâmetro
- Fornecidos ao comando como parte da sequência de caracteres de comando

Para obter uma visão geral dessas opções, consulte "Configurando Opções para Utilização das Interfaces com o Usuário" na página 53. Para obter detalhes integrais dos parâmetros de configuração, consulte o tópico sobre

como configurar o acesso do cliente de linha de comandos no *Tivoli Workload Scheduler: Administration Guide*.

dumpdef *file-path*
Transfere por download o arquivo XML do provedor de eventos genéricos. O arquivo é transferido por download com o nome o caminho de arquivo fornecidos por você em *file-path*. É possível editar o arquivo para incluir seus próprios tipos de evento customizado.

o nome do provedor de eventos genéricos fornecido com o produto é `GenericEventPlugIn`. Você pode alterar seu nome pela tag `name` da palavra-chave `eventPlugin`.

Importante: Você deve utilizar esse nome como o valor:

- Da palavra-chave `source` do comando "sendevent" na página 524
- Da palavra-chave `eventProvider` na definição das regras de eventos emitidas por esses eventos customizados.

loaddef *file-path*
Transfere por upload o arquivo XML modificado do provedor de eventos genéricos a partir do arquivo e caminho fornecidos por você em *file-path*.

Comentários

Os esquemas de linguagem de regra a seguir são utilizados para validar suas definições de evento customizado e, dependendo do editor XML que você possua, para fornecer ajuda de sintaxe:

- eventDefinitions.xsd
- common.xsd

Os arquivos estão localizados no subdiretório `schemas` do diretório de instalação do Tivoli Workload Scheduler.

Ao fazer download do arquivo de modelo do provedor de eventos genéricos, ele é semelhante a este:

```
<?xml version="1.0" encoding="UTF-8"?>
<eventDefinitions
xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/plugins/events"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/
plugins/events/eventDefinitions.xsd" >
<eventPlugin>
<complexName displayName="Custom event" name="GenericEventPlugIn" />
<scopes>
<scope name="Generic">
<scopedef text="{Param1} on {Workstation}" />
</scope>
</scopes>
<!-- Generic Event -->
<event baseAliasName="genericEvt" scope="Generic">
<complexName displayName="Generic event" name="Event1" />
<displayDescription>The event is sent when the specified expression is
matched.</displayDescription>
<property type="string" required="true" wildcardAllowed="true"
multipleFilters="true" minlength="1">
<complexName displayName="Parameter 1" name="Param1" />
<displayDescription>The value of parameter 1</displayDescription>
</property>
<property type="string" required="true" wildcardAllowed="false"
multipleFilters="false" minlength="1>
<complexName displayName="Workstation" name="Workstation" />
<displayDescription>The workstation for which the event is
generated.</displayDescription>
</property>
</property>
</event>
</eventPlugin>
</eventDefinitions>
```

Em seguida, edite esse arquivo para incluir os tipos de propriedades necessárias para definir um evento específico. É possível incluir os seguintes tipos de propriedades:

Tabela 79. Propriedades adicionais que podem ser utilizadas para definir eventos customizados.

Tipo de Propriedade	Inclua no arquivo de evento XML, conforme mostrado
boolean	<property type="boolean" required="false" wildcardAllowed="false" multipleFilters="true" minlength="1"> <complexName displayName="Boolean field" name="Boolean"/> <displayDescription>Add a boolean field</displayDescription> </property>
data	<property type="date" required="false" wildcardAllowed="false" multipleFilters="true" minlength="1"> <complexName displayName="Date field" name="Date"/> <displayDescription>Add a date field</displayDescription> </property>
data/hora	<property type="datetime" required="false" wildcardAllowed="false" multipleFilters="true" minlength="1"> <complexName displayName="Date Time field" name="Datetime"/> <displayDescription>Add a date time field</displayDescription> </property>
datetimeutc	<property type="datetimeutc" required="false" wildcardAllowed="false" multipleFilters="true" minlength="1"> <complexName displayName="Date Time UTC field" name="Datetimeutc"/> <displayDescription>Add a date time UTC field</displayDescription> </property>
estimada	<property type="duration" required="false" wildcardAllowed="false" multipleFilters="true" minlength="1"> <complexName displayName="Duration field" name="Duration"/> <displayDescription>Add a duration field</displayDescription> </property>
fileSize	<property type="fileSize" required="false" wildcardAllowed="false" multipleFilters="true" minlength="1"> <complexName displayName="File size field" name="filesize"/> <displayDescription>Add a file size field</displayDescription> </property>
nonnegativeinteger	<property type="nonnegativeinteger" required="false" wildcardAllowed="false" multipleFilters="true" minlength="1"> <complexName displayName="Non negativeinteger field" name="nonnegativeinteger"/> <displayDescription>Add a non negativeinteger field</displayDescription> </property>
numeric	<property type="numeric" required="false" wildcardAllowed="false" multipleFilters="true" minlength="1"> <property type="numeric" required="false" wildcardAllowed="false" multipleFilters="true" minlength="1"> <displayDescription>Add a numeric field</displayDescription> </property>
porcentagem	<property type="percentage" required="false" wildcardAllowed="false" multipleFilters="true" minlength="1"> <complexName displayName="Percentage field" name="percentage"/> <displayDescription>Add a percentage field</displayDescription> </property>
cadeia	<property type="string" required="true" wildcardAllowed="true" multipleFilters="true" minlength="1"> <complexName displayName="String with wildcards" name="StringWithWildcards"/> <displayDescription>Add a string with wildcards</displayDescription> </property> or <property type="string" required="true" wildcardAllowed="false" multipleFilters="true" minlength="1"> <complexName displayName="String without wildcards" name="StringWithoutWildcards"/> <displayDescription>Add a string without wildcards</displayDescription> </property>
horário	<property type="time" required="false" wildcardAllowed="false" multipleFilters="true" minlength="1"> <complexName displayName="Time field" name="Time"/> <displayDescription>Add a time field</displayDescription> </property>

É possível alterar os valores de todos os atributos da propriedade, com exceção do `tipo`, para atender seus requisitos.

As propriedades definidas são convertidas em campos de entrada depois que a definição de evento é transferida por upload e aberta no Dynamic Workload Console.

Também é possível definir mais de um evento repetindo as seções `<eventPlugin>...</eventPlugin>`. Por exemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<eventDefinitions
xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/plugins/events"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/
plugins/events/eventDefinitions.xsd" >
<eventPlugin>
<complexName displayName="Custom event" name="GenericEventPlugIn" />
<scopes>
<scope name="Art042StockQuantity">
<scopedef text="{Art042pieces}"/>
</scope>
</scopes>
<!-- Generic Event -->
<event baseAliasName="genericEvt" scope="Generic">
<complexName displayName="Stock of article 042 reaches minimum level" name="art042qty"/>
<displayDescription>O evento é enviado quando o número de itens do art.42 do estoque atinge um
nível mínimo.</displayDescription>
<property type="numeric" required="true" wildcardAllowed="false"
multipleFilters="true" minlength="1">
<complexName displayName="Art042 items on stock" name="art042items"/>
<displayDescription>The number of art042 items left</displayDescription>
</property>
</event>
</eventPlugin>
<eventPlugin>
<complexName displayName="Custom event" name="GenericEventPlugIn" />
<scopes>
<scope name="HDAlmostFull">
<scopedef text="{HDNearingFullPercent} on {Workstation}"/>
</scope>
</scopes>
<!-- Generic Event 2-->
<event baseAliasName="Hard drive saturation" scope="Generic">
<complexName displayName="Hard drive saturation" name="HDSatEvent"/>
<displayDescription>displayDescription>The event is sent when the percentage field
reaches the warning level.</displayDescription>
<property type="percentage" required="true" wildcardAllowed="false"
multipleFilters="true" minlength="1>
<complexName displayName="Percentage Full" name="PercentFull"/>
<displayDescription>The percentage of total disk space used</displayDescription>
</property>
<property type="string" required="true" wildcardAllowed="false"
multipleFilters="false" minlength="1>
<complexName displayName="Workstation" name="Workstation" />
<displayDescription>The workstation where the hard drive is installed</displayDescription>
</property>
</event>
</eventPlugin>
</eventDefinitions>
```

Exemplos

Neste exemplo, você:

1. Faça download do arquivo XML do provedor de eventos genéricos como o arquivo `c:\custom\myevents.xml`

   ```
   evtdef dumpdef c:\custom\myevents.xml
   ```

2. Edita o arquivo para incluir suas próprias definições de tipo de evento.
3. Ao concluir, transfira por upload o arquivo XML do provedor de eventos genéricos a partir do arquivo `c:\custom\myevents.xml`

   ```
   evtdef loaddef c:\custom\myevents.xml
   ```

evtsize

Define o tamanho dos arquivos de mensagens do Tivoli Workload Scheduler. Este comando é usado pelo administrador do Tivoli Workload Scheduler para aumentar o tamanho de um arquivo de mensagens após o recebimento da mensagem, "Fim de arquivo no arquivo de eventos.", ou para monitorar o tamanho da fila de mensagens contida no arquivo de mensagens. Você deve ser **maestro** ou **root** no UNIX ou **Administrator** no Windows para executar **evtsize**. Pare o mecanismo do IBM Tivoli Workload Scheduler antes de executar esse comando.

Sintaxe

evtsize -V | -U

evtsize *filename size*

evtsize -compact *filename* [*size*]

evtsize -info *filename*

evtsize -show *filename*

evtsize -info | -show pobox

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

-compact *filename* **[***size***]**
Reduz o tamanho do arquivo de mensagens especificado para o tamanho ocupado pelas mensagens presentes no momento que o comando é executado. É possível usar como opção essa palavra-chave para também especificar um novo tamanho de arquivo.

-info*filename*
Exibe o uso da porcentagem da fila de mensagens contida no arquivo de mensagens.

-show *filename*
Exibe o tamanho da fila de mensagens contida no arquivo de mensagem

nome do arquivo
O nome do arquivo de evento. Especifique um dos seguintes:

```
Courier.msg
Intercom.msg
Mailbox.msg
Planbox.msg
pobox/workstation.msg
mirrorbox.msg
```

tamanho
O tamanho máximo do arquivo de evento em bytes. Não deve ser menos que 1048576 bytes (1 MB).

Quando construído pela primeira vez pelo Tivoli Workload Scheduler, o tamanho máximo é definido para 10 MB.

Nota: O tamanho do arquivo de mensagens é igual ou maior do que o tamanho real da fila de mensagens contida e aumenta progressivamente até que a fila de mensagens fique vazia; quando isso ocorre, o arquivo de mensagens é esvaziado.

-info | -show pobox
Exibe o nome do arquivo de mensagens no diretório `pobox`, com o maior tamanho de fila calculado como uma porcentagem do tamanho total do arquivo. O nome do arquivo e a porcentagem usados são retornados. **-info** e **-show** retornam os mesmos resultados.

Exemplos

Para definir o tamanho máximo do arquivo `Intercom.msg` para 20 MB, execute o seguinte comando:

```
evtsize Intercom.msg 20000000
```

Para definir o tamanho máximo do arquivo `pobox` da estação de trabalho `chicago` para 15 MB, execute o seguinte comando:

```
evtsize pobox\chicago.msg 15000000
```

O seguinte comando:

```
evtsize -show Intercom.msg
```

retorna a seguinte saída:

```
Tivoli Workload Scheduler (UNIX)/EVTSIZE 8.3 (1.2.2.4) Materiais Licenciados -
Propriedade da IBM(R)
5698-WSH
(C) Copyright da IBM Corp 1998, 2006 Todos os direitos reservados.
Direitos Restritos para Usuários do Governo dos Estados Unidos
Uso, duplicação ou divulgação restritos pelo documento
GSA ADP Schedule Contract com a IBM Corp.
IBM é uma marca registrada da International Business Machines
Corporation nos Estados Unidos e/ou em outros países.
AWSDEK703I Tamanho da fila atual 240, número máximo de bytes 10000000 (leitura 48, gravação 288)
```

em que:

880 É o tamanho da fila atual do arquivo `Intercom.msg`

10000000
É o tamanho máximo do arquivo `Intercom.msg`

read 48
É posição do ponteiro para ler registros.

write 928
É posição do ponteiro para gravar registros.

Se o comando a seguir:

```
evtsize -info Mailbox.msg
```

retornar:

```
25
```

significa que 25 por centro do arquivo foi usado.

jobinfo

Utilizado em um script de job para retornar informações sobre o job. Este comando não é suportado em agentes dinâmicos, conjuntos, conjuntos dinâmicos e tipos de tarefas com opções avançadas.

Sintaxe

jobinfo -V | -U

jobinfo *job-option* [...]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

job-option

A opção de job. Especifica um ou mais dos seguintes:

confirm_job
Retornará **YES** se o job requerer confirmação.

is_command
Retornará **YES** se o job tiver sido planejado ou submetido com a construção **docommand**.

job_name
Retorna o nome do job sem os nomes da estação de trabalho e do fluxo de job.

job_pri
Retorna o nível de prioridade do job.

programmatic_job
Retorna **YES** se a tarefa tiver sido enviada com a utilização do comando **at** ou **batch**. Somente para o UNIX.

re_job Retornará **YES** se a tarefa estiver sendo reexecutada como resultado de um comando **rerun** do **conman**, ou da opção de recuperação de reexecução.

re_type
Retornará a opção de recuperação do job (**stop**, **continue** ou **rerun**).

rstrt_flag
Retornará **YES** se o job estiver sendo executado como o job de recuperação.

rstrt_retcode
Se o job atual for um job de recuperação, retornará o código de retorno do job pai.

planejamento
Retorna o nome do fluxo de tarefas no qual a tarefa é enviada.

schedule_ia
Retorna a hora e data que o fluxo de tarefas está planejado para iniciar.

schedule_id
Retorna o *jobstream_ID* do fluxo de tarefas no qual a tarefa é enviada.

time_started
Retorna a hora em que o job iniciou sua execução.

Comentários

Os valores de opção do job são retornados, um em cada linha, na mesma ordem em que foram solicitados.

Exemplos

1. O arquivo de script `/jcl/backup` é referenciado duas vezes, fornecendo-lhe os nomes de job **partback** e **fullback**. Se o job for executado como **partback**, ele executará um backup parcial. Se for executado como **fullback**, executará um backup completo. No script, os comandos como os seguintes são utilizados para fazer a distinção:

   ```
   #Determinar parcial (1) ou completo (2):
   if [ "`\`maestro\`/bin/jobinfo job_name`" = "PARTBACK" ]
   then
   bkup=1
   else
   bkup=2
   fi
   ...
   ```

2. Para exibir o código de retorno do job pai, se o job atual for um job de recuperação, execute o seguinte comando:

   ```
   $ jobinfo rstrt_retcode
   ```

   O primeiro job (job pai) foi definido no script `recovery.sh`, enquanto que o segundo (job de recuperação) será ativado apenas se o primeiro terminar de forma anormal.

   Quando combinados com uma condição de código de retorno, **jobinfo rstrt_retcode** pode ser utilizado para direcionar o job de recuperação para

executar ações diferentes, dependendo do código de retorno do job pai. O exemplo a seguir mostra um job de recuperação:

```
$JOBS
MASTER#DBSELOAD DOCOMMAND "/usr/local/tws/maestro/scripts/populate.sh"
STREAMLOGON "^TWSUSER^"
DESCRIPTION "populate database manual"
RECOVERY RERUN AFTER MASTER#RECOVERY
RCCONDSUCC "(RC = 0) OR ((RC > 4) AND (RC < 11))"
```

Nota: A tarefa é definida com a ação de recuperação RERUN. Isso permite que o job de recuperação execute ações corretivas antes que o job pai tente executar novamente.

O próprio job de recuperação é definido conforme mostrado no exemplo a seguir:

```
$ JOBS
MASTER#RECOVERY DOCOMMAND "^TWSHOME^/scripts/recovery.sh"
STREAMLOGON "^TWSUSER^"
DESCRIPTION "populate database recovery manual"
RECOVERY STOP
```

jobstdl

Retorna os nomes de arquivos de lista padrão. Esse comando deve ser executado pelo usuário para o qual o Tivoli Workload Scheduler foi instalado. Se você utilizar esse comando sem parâmetros, assegure-se de ter feito logon como usuário do Tivoli Workload Scheduler.

Sintaxe

jobstdl -V | -U

jobstdl
[**-day** *num*]
[{**-first** | **-last** | **-num** *n* | **-all**}]
[**-twslog**]
[{**-name** ["*jobstreamname* [(*hhmm date*),(*jobstream_id*)].]*jobname*"
| *jobnum* |
-schedid *jobstream_id.jobname*}]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

-day *num*
Retorna os nomes de arquivos de lista padrão que são os números especificados de dias anteriores (1 para ontem, 2 para anteontem, e assim por diante). O padrão é zero (hoje).

-first Retorna o nome do primeiro arquivo de lista padrão qualificado.

-last Retorna o nome do último arquivo de lista padrão qualificado.

-num *n*
Retorna o nome do arquivo de lista padrão da execução especificada de um job.

-all Retorna o nome de todos os arquivos de lista padrão qualificados.

-twslog
Retorna o caminho do arquivo *stdlist* do dia atual.

-name **["***jobstreamname***[(***hhmm date***), (***jobstream_id***)].]***jobname***"** | *jobnum*
Especifica a instância do fluxo de tarefas e o nome da tarefa para os quais nomes de arquivos de lista padrão são retornados.

jobnum
Especifica o número do job para o qual os nomes do arquivo de lista padrão são retornados.

-schedid *jobstream_id.jobname*
Especifica o ID do fluxo de tarefas e o nome da tarefa para a qual os nomes de arquivo da lista padrão são retornados.

Comentários

Os nomes de arquivos são retornados em um formato adequado para entrada de outros comandos. São retornados vários nomes separados por um espaço.

Ao usar a sintaxe integral do argumento **-name**, os colchetes na expressão `[(hhmm date), (jobstream_id)]` fazem parte do comando, não são indicadores de sintaxe. Além disso, toda a sequência de identificação da tarefa deve estar entre aspas duplas se a parte que identifica a instância do fluxo de tarefas contiver espaços em branco. Por exemplo, como o *schedtime*, representado por `hhmm date`, possui um espaço nele, você deve colocar a identificação da tarefa inteira entre aspas duplas.

Também é possível executar versões abreviadas do argumento **-name** usando uma sintaxe mais simples. Se quiser saídas menos específicas do comando, é possível especificar apenas *schedtime* (a *data* não é necessária se for para o mesmo dia) ou *jobstream_id* juntamente com *jobname*. Desde que não haja espaços em branco nos argumentos, é possível omitir as aspas duplas. Também é possível omitir os colchetes se você não especificar *schedtime* e *jobstream_id*.

Os exemplos a seguir mostram a sintaxe que você deve usar com o argumento **-name** para os diferentes tipos de informações que espera em retorno, desde as mais específicas até as mais gerais. No exemplo, `job_stream1` é o nome do fluxo de tarefas, `0600 04/05/06` é a hora planejada, `0AAAAAAAAAAAAAB5` é o ID do fluxo de tarefas e `job1` é o nome da tarefa. O número de tarefa de `job1` é `310`. É possível executar **jobstdl** para `job1` da seguinte forma:

```
jobstdl -name "job_stream1[(0600 04/05/10),(0AAAAAAAAAAAAAB5)].job1"
```

Retorna o nome do arquivo de lista padrão `job1` para a instância específica `job_stream1` com *schedtime* e *jobstream_id* especificados.

```
jobstdl -name job_stream1(0AAAAAAAAAAAAAB5).job1
```

Retorna o nome do arquivo de lista padrão para `job1` para a instância `job_stream1` com ID `0AAAAAAAAAAAAAB5`.

```
jobstdl -name "job_stream1(0600 04/05/10).job1"
```

Retorna os nomes de arquivos de lista padrão para `job1` para todas as instâncias possíveis de `job_stream1` programadas para execução às 6h de 05/04/10.

```
jobstdl -name job_stream1(0600).job1
```

Retorna os nomes de arquivos de lista padrão para `job1` para todas as instâncias possíveis de `job_stream1` programadas para execução às 6h do dia atual.

```
jobstdl -name 310
```

Retorna os nomes de arquivos da lista padrão para `job1` para todas as instâncias que tinham número de tarefa `310`.

Exemplos

Para retornar os nomes de caminhos de todos os arquivos de lista padrão do dia atual, execute o seguinte comando:

```
jobstdl
```

Para retornar o nome do caminho da lista padrão para a primeira execução da tarefa `MY_CPU#ELI[(1824 03/09/06),(0AAAAAAAAAAAAAEE)].DIR` no dia atual, execute o seguinte comando:

```
jobstdl -first -name "MY_CPU#ELI[(1824 03/09/06),(0AAAAAAAAAAAAAEE)].DIR"
```

Para retornar o nome do caminho da lista padrão para a primeira execução da tarefa `0AAAAAAAAAAAAAEE.DIR` no dia atual, execute o seguinte comando:

```
jobstdl -first -schedid 0AAAAAAAAAAAAAEE.DIR
```

Para retornar o nome do caminho da lista padrão para a segunda execução da tarefa `MY_CPU#ELI[(1824 03/09/06),(0AAAAAAAAAAAAAEE)].DIR` no dia atual, execute o seguinte comando:

```
jobstdl -num 2 -name "MY_CPU#ELI[(1824 03/09/06),(0AAAAAAAAAAAAAEE)].DIR"
```

Para retornar os nomes de caminho dos arquivos de lista padrão para todas as execuções da tarefa `MY_CPU#ELI[(1824 03/09/06),(0AAAAAAAAAAAAAEE)].DIR` de três dias atrás, execute o seguinte comando:

```
jobstdl -day 3 -name "MY_CPU#ELI[(1824 03/09/06),(0AAAAAAAAAAAAAEE)].DIR"
```

Para retornar o nome do caminho da lista padrão para a última execução da tarefa `MY_CPU#ELI[(1824 03/09/06),(0AAAAAAAAAAAAAEE)].DIR` de quatro dias atrás, execute o seguinte comando:

```
jobstdl -day 4 -last -name "MY_CPU#ELI[(1824 03/09/06),(0AAAAAAAAAAAAAEE)].DIR"
```

Para retornar o nome do caminho da lista padrão do job número **455**, execute o seguinte comando:

```
jobstdl 455
```

Para imprimir o conteúdo do arquivo de lista padrão do job número **455**, execute o seguinte comando:

```
cd `maestro`/bin
lp -p 6 `jobstdl 455`
```

maestro

Retorna o nome do caminho do diretório home do Tivoli Worload Scheduler, referido como *TWS_home*.

Sintaxe

maestro [**-V** | **-U**]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

Exemplos

Para exibir o diretório home do Tivoli Workload Scheduler, execute o seguinte comando:

```
$ maestro
/usr/lib/maestro
```

Para alterar o diretório para o diretório home do Tivoli Workload Scheduler, execute o seguinte comando:

```
$ cd `maestro`
```

makecal

Cria um calendário personalizado. No UNIX, o shell Korn é necessário para executar esse comando.

Sintaxe

makecal [**-V** | **-U**]

makecal
[**-c** *name*]
-d *n*
| **-e**
| {**-f 1** | **2** | **3 -s** *date*}
| **-l**
| **-m**
| **-p** *n*
| {**-r** *n* **-s** *date*}
| **-w** *n*
[**-i** *n*]
[**-x** | **-z**]
[**-freedays** *Calendar_Name* [**-sa**] [**-su**]]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

-c *name*
Especifica um nome para o calendário. Palavras-chave do Tivoli Workload Scheduler (como *Freedays* ou *Schedule*) não podem ser utilizadas como nomes de calendário. O nome pode conter até oito caracteres alfanuméricos e deve iniciar com uma letra. Não utilize os nomes de dias da semana para nomes de calendário. O nome padrão é: **C***hhmm*, em que *hhmm* é a hora e minutos atuais.

-d *n* Especifica o e*n*ésimo dia de cada mês.

-e Especifica o último dia de cada mês.

-f 1 | 2 | 3
Cria um calendário fiscal de final de mês contendo o último dia do mês fiscal. Especifique um dos seguintes formatos:

1 formato de semana 4-4-5

2 formato de semana 4-5-4

3 formato de semana 5-4-4

Esse argumento requer o argumento **-s**.

-i *n* Especifica para inserir *n* datas no calendário.

-l Especifica o último dia útil de cada mês. Para esse argumento funcionar adequadamente, o plano de produção (arquivo Symphony) e o calendário **holidays** já deverão existir.

Nota: O uso desse argumento resulta na inclusão pelo novo calendário do último dia útil do mês anterior à data de criação do calendário.

-m Especifica o primeiro e o décimo quinto dias de cada mês.

-p *n* Especifica o dia útil antes do e*n*ésimo dia de cada mês. Para esse argumento funcionar adequadamente, o plano de produção (arquivo Symphony) e o calendário **holidays** já deverão existir

-r *n* Especifica todo e*n*ésimo dia. Esse argumento requer o argumento **-s**.

-s *date* Especifica a data de início para os argumentos **-f** e **-r**. A data deve ser colocada entre aspas e deve ser válida e não ambígua, por exemplo, utilize **JAN 10 2005**, não **1/10/05**. Consulte *base-date* para **datecalc** na página 500 para obter mais informações sobre os formatos de datas.

-w *n* Especifica o dia útil após o e*n*ésimo dia do mês. Para esse argumento funcionar adequadamente, o plano de produção (arquivo Symphony) e o calendário **holidays** já deverão existir.

-x Envia a saída do calendário para **stdout** em vez de inclui-la no banco de dados.

-z Inclui o calendário no banco de dados e compila o plano de produção (arquivo Symphony).

Nota: Esse argumento reenvia tarefas e fluxos de tarefas do plano de produção do dia atual. Talvez seja necessário cancelar fluxos de job e jobs.

-freedays
Especifica o nome de um calendário de dias livres *Nome_do_Calendário* que deve substituir **feriados** na avaliação de *dias úteis*.

Nesse caso, *dias úteis* é avaliado como *todos os dias* excluindo *sábado*, *domingo* e todas as datas listadas em *Nome_do_Calendário*.

Por padrão, *sábado* e *domingo* não são considerados *dias úteis*, a não ser que seja explicitamente especificado de outra forma incluindo-se **-sa** e/ou **-su** depois de *Calendar_Name* .

Também é possível especificar **feriados** como o nome do calendário de dias livres.

Essa palavra-chave afeta o processamento de **makecal** com as opções **-l**, **-p** e **-w**.

Exemplos

Para criar um calendário de dois anos com o último dia de cada mês selecionado, execute o seguinte comando:

```
makecal -e -i 24
```

Para criar um calendário com 30 dias que começa em 30 de maio de 2005 e com cada terceiro dia selecionado, execute o seguinte comando:

```
makecal -r 3 -s "30 MAY 2005" -i 30
```

metronome

Metronome é substituído por **tws_inst_pull_info**. Consulte o *Guia de Resolução de Problemas do IBM Tivoli Workload Scheduler* para obter informações sobre este comando.

morestdl

Exibe o conteúdo de arquivos de lista padrão. Esse comando deve ser executado pelo usuário para o qual o Tivoli Workload Scheduler foi instalado. Se você utilizar esse comando sem parâmetros, assegure-se de ter feito logon como usuário do Tivoli Workload Scheduler. Este comando é suportado para agentes tolerantes a falha e agentes padrão.

Sintaxe

morestdl -V | -U

morestdl
[**-day** *num*]
[**-first** | **-last** | **-num** *n* | **-all**]
[**-twslog**]
[{**-name** ["*jobstreamname* [(*hhmm date*),(*jobstream_id*)].]*jobname*"
| *jobnum* |
-schedid *jobstream_id.jobname*}]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

-day *num*
Exibe arquivos de lista padrão que são os números especificados de dias anteriores (1 para ontem, 2 para anteontem, e assim por diante). O padrão é zero (hoje).

-first Exibe o primeiro arquivo de lista padrão qualificado.

-last Exibe o último arquivo de lista padrão qualificado.

-num *n*
Exibe o arquivo de lista padrão da execução especificada de um job.

-all Exibe todos os arquivos de lista padrão qualificados.

-twslog
Exibe o conteúdo do arquivo *stdlist* do dia atual.

-name ["*jobstreamname* **[(***hhmm date***),(***jobstream_id***)].]***jobname***" | ***jobnum*
Especifica a instância do fluxo de tarefas e o nome da tarefa para os quais o arquivo de lista padrão é exibido.

jobnum
Especifica o número do job para o qual o arquivo de lista padrão é exibido.

-schedid *jobstream_id.jobname*
Especifica o ID do fluxo de tarefas e o nome da tarefa para a qual os nomes de arquivo da lista padrão são retornados.

Comentários

Os colchetes na expressão [(*hhmm date*), (*jobstream_id*)] fazem parte do comando, não são indicadores de sintaxe. Isto significa que é possível fornecer qualquer um dos seguintes para o argumento **-name**:

```
morestdl -name ["jobstreamname[(hhmm date),(jobstream_id)].jobname"
morestdl -name jobnum
```

Toda a sequência de identificação da tarefa deve estar entre aspas duplas se a parte que identifica a instância do fluxo de tarefas contiver espaços em branco. Por exemplo, como o *schedtime*, representado por *hhmm date*, possui um espaço nele, você deve colocar a identificação da tarefa inteira entre aspas duplas.

Se você apenas desejar identificar um nome de tarefa, não precisará das aspas duplas.

A seguir está um exemplo da sintaxe para utilização ao identificar uma tarefa com e sem seu fluxo de tarefas. No exemplo, `job_stream1` é o nome do fluxo de tarefas, `0600 04/05/06` é a hora planejada, `0AAAAAAAAAAAAAB5` é o ID do fluxo de tarefas e `job1` é o nome da tarefa. É possível executar o comando **morestdl** em `job1` utilizando um destes dois formatos:

```
morestdl -name "job_stream1[(0600 04/05/06),(0AAAAAAAAAAAAB5)].job1"
morestdl -name job1
```

Exemplos

Para exibir o arquivo de lista padrão para a primeira execução da tarefa `MY_CPU#ELI[(1824 03/09/06),(0AAAAAAAAAAAAAEE)].DIR` no dia atual, execute o seguinte comando:

```
morestdl -first -name "MY_CPU#ELI[(1824 03/09/06),(0AAAAAAAAAAAAEE)].DIR"
```

Para exibir o arquivo da lista padrão para a primeira execução da tarefa `0AAAAAAAAAAAAAEE.DIR` no dia atual, execute o seguinte comando:

```
morestdl -first -schedid 0AAAAAAAAAAAAEE.DIR
```

Para exibir o arquivo de lista padrão para a segunda execução da tarefa `MY_CPU#ELI[(1824 03/09/06),(0AAAAAAAAAAAAAEE)].DIR` no dia atual, execute o seguinte comando:

```
morestdl -num 2 -name "MY_CPU#ELI[(1824 03/09/06),(0AAAAAAAAAAAAEE)].DIR"
```

Para exibir os arquivos de lista padrão para todas as execuções da tarefa `MY_CPU#ELI[(1824 03/09/06),(0AAAAAAAAAAAAAEE)].DIR` de três dias atrás, execute o seguinte comando:

```
morestdl -day 3 -name "MY_CPU#ELI[(1824 03/09/06),(0AAAAAAAAAAAAEE)].DIR"
```

Para exibir o arquivo de lista padrão para a última execução da tarefa `MY_CPU#ELI[(1824 03/09/06),(0AAAAAAAAAAAAAEE)].DIR` de quatro dias atrás, execute o seguinte comando:

```
morestdl -day 4 -last -name "MY_CPU#ELI[(1824 03/09/06),(0AAAAAAAAAAAAEE)].DIR"
```

Para imprimir o arquivo de lista padrão do job número 455, execute o seguinte comando:

```
morestdl 455 | lp -p 6
```

parms

Gerencia parâmetros definidos localmente nas estações de trabalho. Os parâmetros gerenciados por **parms** podem ser utilizados somente em definições de tarefas ou de fluxos de tarefas com as palavras-chave **scriptname** ou **opens** ou em um arquivo de script de tarefa.

Esses parâmetros são resolvidos no momento do envio na estação de trabalho em que a tarefa ou o fluxo de tarefas é enviado. Se não houver correspondência entre o*parametername* especificado e o nome dos parâmetros definidos no banco de dados local na estação de trabalho, um valor *null* será retornado.

Autorização

Você deve ter acesso ***display*** para o banco de dados de parâmetros definido localmente. Além disso, você deve estar autorizado com o seguinte acesso:

build on object file
Se você utilizar a opção **-b** para criar ou reconstruir o banco de dados de parâmetros local.

excluir
Se você utilizar a opção **-d** para excluir as definições de parâmetros.

modify on object file
Se você utilizar a opção **-replace** para incluir ou modificar as definições de parâmetros.

Sintaxe

parms {[**-V** | **-u**] | **-build**}

parms {**-replace** | **-extract**} *filename*

parms [**-d**]*parametername*

parms -c *parametername value*

Argumentos

-V Exibe a versão do comando e sai.

-u Exibe informações de utilização do comando e sai.

-build Cria o banco de dados de parâmetros na estação de trabalho se ele não existir. Reconstrói o banco de dados de parâmetros, removendo os registros não utilizados e evitando a fragmentação de várias inclusões e exclusões, se ele já existir.

-extract
Extrai todas as definições de parâmetros do banco de dados local e as armazena no arquivo com o nome *filename*. Utilize esta opção se desejar exportar definições de parâmetros locais para importá-las como definições de parâmetros globais no banco de dados de objetos de planejamento utilizando os comandos “add” na página 287 ou “replace” na página 324.

-replace
Incluir no banco de dados local novas definições de parâmetros armazenadas em um arquivo denominado*filename* ou substituir as já existentes. Utilize esta opção se você desejar importar, como definições de

parâmetros locais, as definições de parâmetros globais contidas no arquivo denominado*filename* e extraídas do banco de dados de objetos de planejamento utilizando o comando "extract" na página 300.

-d Exclui os parâmetros com nome *parametername* do banco de dados local na estação de trabalho.

parametername
Especifica o nome do parâmetro cujo valor é exibido. Quando utilizado com o argumento **-d**, representa o nome do parâmetro a ser excluído.

-c *name value*
Especifica o nome e o valor de um parâmetro. O nome pode conter até 16 caracteres alfanuméricos, incluindo traços (-) e sublinhados (_), e deve iniciar com uma letra. O valor pode conter até 72 caracteres. Coloque o valor entre aspas duplas se ele contiver caracteres especiais. Se o parâmetro não existir, ele será adicionado ao banco de dados. Se o parâmetro já existir, seu valor será alterado.

Comentários

Quando**parms** for executado na linha de comandos sem argumentos, ele solicitará nomes e valores de parâmetros.

A utilização de **parms** em definições de tarefa e em arquivos de script de tarefa requer que o parâmetro já exista localmente no banco de dados de parâmetros na estação de trabalho.

Este é um uso de amostra de um parâmetro local, `MYFILE`, em uma cláusula de dependência de arquivo:

```
schedule test_js
on everyday
opens "/usr/home/tws_99/'/usr/home/tws_99/bin/parms MYFILE'"
:
test_job
end
```

O exemplo a seguir explica como a variável ***var*** colocada entre circunflexos (^) é substituída enquanto a tarefa está em processo. Se a tarefa for enviada como uma tarefa ad hoc, o parâmetro***var*** é expandido, ou seja, substituído pelo valor designado a ***var*** no banco de dados local, no tempo de envio e não quando a tarefa é ativada.

Exemplo de definição de tarefa do UNIX:

```
DATA#UX_P_TEST DOCOMMAND "ls ^var^"
STREAMLOGON "mae82"
DESCRIPTION "Test parms in job definition on UNIX."
RECOVERY STOP
```

Exemplo de definição de tarefa do Windows:

```
BORG#WIN_P_TEST DOCOMMAND "dir ^var^"
STREAMLOGON "mae82"
DESCRIPTION "Test parms in job definition on Windows."
RECOVERY STOP
```

Quando utilizado em um arquivo de script da tarefa, o parâmetro não é expandido até o script ser ativado. Não é expandido quando o fluxo de tarefa que contém a tarefa é processado por **JnextPlan**. Esses são exemplos de como utilizar o parâmetro***var*** nos arquivos de script da tarefa.

Exemplo de script do UNIX:

```
#!/bin/sh
TWS_HOME="/opt./tws/mae82/maestro"
export TWS_HOME
MDIR='$TWS_HOME/bin/parms var'
export MDIR
ls -l $MDIR
```

Exemplo de script do Windows:

```
set TWS_HOME=d:\win32app\TWS\mae82\maestro
echo %TWS_HOME%
FOR /F "Tokens=*" %%a in (%TWS_HOME%\bin\parms var) do set MDIR=%%a
echo %MDIR%
dir %MDIR%
```

Exemplos

Para retornar o valor de `myparm`, execute o seguinte comando:

```
parms myparm
```

Para alterar o valor de `myparm`, execute o seguinte comando:

```
parms -c myparm "item 123"
```

Para criar um novo parâmetro chamado`hisparm`, execute o seguinte comando:

```
parms -c hisparm "item 789"
```

Para alterar o valor de `myparm` e incluir `herparm`, execute o seguinte comando:

```
parms
Nome do parâmetro ? myparm < Return>
Valor do parâmetro? "item 456" < Return>
Nome do parâmetro ? herparm < Return>
Valor do parâmetro? "item 123" < Return>
Nome do parâmetro ? < Return>
```

Para obter informações adicionais, consulte a Capítulo 6, “Customizando a Carga de Trabalho Utilizando Tabelas de Variáveis”, na página 113.

release

Libera tarefas e fluxos de tarefas das dependências **needs** em um recurso. Este comando deve ser emitido somente de dentro do arquivo de script da tarefa.

Sintaxe

release -V | -U

release
[**-s**]
[*workstation***#**]
resourcename
[*count*]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

-s Libera a dependência **needs** do recurso especificado somente no nível do fluxo de tarefas.

Se **-s** não for utilizado, a dependência **needs** do recurso especificado é liberada no nível da tarefa ou no nível do fluxo de tarefas se a dependência **needs** desse recurso não for localizada no nível da tarefa.

workstation **#**

Especifica o nome da estação de trabalho ou classe de estação de trabalho na qual o recurso está definido. O padrão é a estação de trabalho local.

resourcename

Especifica o nome do recurso envolvido na dependência **needs**.

count Especifica o número de unidades do recurso a ser liberado.

Comentários

Unidades de um recurso são adquiridas por um job ou fluxo de job na hora em que é lançado e são liberadas automaticamente quando o job ou fluxo de job é concluído. O comando **release** pode ser utilizado em um script de tarefa para liberar recursos antes da conclusão da tarefa ou do fluxo de tarefas ou para liberar manualmente tarefas e fluxos de tarefas das dependências needs nas situações de emergência.

Exemplos

No fluxo de tarefas a seguir, duas unidades do recurso`dbase` são requeridas pelo fluxo de tarefas `sked5`:

```
schedule ux1#sked5 on tu
needs 2 dbase :
job1
jobrel follows job1
job2 follows jobrel
end
```

Para liberar o recurso`dbase` antes de `job2` ser iniciado, o arquivo de script para `jobrel` contém o seguinte comando:

Em sistemas operacionais UNIX:

```
`maestro`/bin/release -s dbase
```

Nos sistemas operacionaisWindows:

```
<TWS_home>\bin\release -s dbase
```

Nota: O argumento **-s** pode ser omitido, porque não foram reservados recursos no nível do job.

rmstdlist

Remove ou exibe arquivos de lista padrão com base no tempo do arquivo. Esse utilitário deve ser utilizado pelo administrador do Tivoli Workload Scheduler para manter o ambiente de planejamento.

Sintaxe

rmstdlist -V | -U

rmstdlist [**-p**] [*age*]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

-p Exibe os nomes dos diretórios de arquivos de lista padrão qualificados. Nenhum diretório ou arquivo é removido. Se você não especificar **-p**, os arquivos de lista padrão qualificados serão removidos.

age O tempo mínimo, em dias, para os diretórios de arquivos de lista padrão serem exibidos ou removidos. O padrão é 10 dias.

Nota: Como a lista de diretórios e arquivos mostrados ou excluídos utilizando**rmstdlist** é produzida com base na última vez em que foram acessados, as datas mostradas na lista de diretórios poderia ser diferente das datas exibidas na lista de arquivos.

Sintaxe

Como regra, você deve remover regularmente os arquivos da lista padrão de algum lugar a cada 10-20 dias. As listas não processadas maiores podem ser difíceis de gerenciar e, se o número de arquivos se tornar excessivamente grande, poderá ser necessário apagar alguns deles manualmente antes de poder utilizar **rmstdlist** novamente.

Este problema pode ocorrer nos sistemas AIX, particularmente devido a uma limitação atualmente não resolvida com o comando `rm -rf`. Quando**rmstdlist** falhar devido a esta limitação, ele não exibirá nenhum erro diferente do código de saída 126. Se você, em vez disso, tiver o error`rm -rf` exibido, poderá editar o script `rmstdlist` da seguinte maneira:

1. Localize o script no diretório *TWS_home*`/bin`
2. Localize a linha:

   ```
   rm -rf `cat /tmp/rm$$` 2> /dev/null
   ```

3. Remova a redireção para `/dev/null` para que a linha se torne:

   ```
   rm -rf `cat /tmp/rm$$`
   ```

Exemplos

Para exibir os nomes dos diretórios do arquivo de lista padrão com mais de 14 dias, execute o seguinte comando:

```
rmstdlist -p 14
```

Para remover todos os arquivos de lista padrão (e seus diretórios) com mais de sete dias, execute o seguinte comando:

```
rmstdlist 7
```

sendevent

O comando envia os eventos customizados definidos com o comando evtdef para o servidor do processador de eventos atualmente ativo no plano de produção. À medida que os eventos são recebidos pelo processador de eventos, acionam as regras de eventos nas quais foram especificados.

Os usuários podem substituir o servidor de destino padrão (definido pelas opções globais) especificando o host e a porta de um novo servidor.

Sintaxe

sendevent -V | ? | -help | -u | -usage

sendevent [**-hostname** *hostname*]
[{**-port | -sslport**} *port*]

eventType
source
[[*attribute=value*]...]

Argumentos

-V Exibe a versão do comando e sai.

? | -help | -u | -usage
Exibe informações de utilização do comando e sai.

-hostname *hostname*
Especifica o nome do host de um servidor do processador de evento alternativo diferente do atualmente ativo. Esse parâmetro é necessário se o comando for ativado a partir de um cliente de linha de comandos.

-port | -sslport} *port*
Especifica o número da porta de um servidor do processador de eventos alternativo diferente do atualmente ativo. **-sslport** define a porta usada para receber conexões SSL de entrada. Esse parâmetro é necessário se o comando for ativado a partir de um cliente de linha de comandos.

eventType
Um dos tipos de evento customizado, definido com o comando evtdef no provedor de eventos genéricos e especificado como o evento acionador em uma definição de regra de evento.

origem O nome do provedor de eventos que você customizou com evtdef. Esse também é o nome que você deve especificar como o argumento para a palavra-chave `eventProvider` na definição das regras de evento acionadas por esses eventos customizados.

O nome padrão é `GenericEventPlugIn`.

attribute=value
Um ou mais dos atributos qualificadores do tipo de evento customizado que estão especificados como atributos do evento acionador para a regra de evento.

Comentários

Este comando também pode ser executado em sistemas nos quais apenas o cliente da linha de comando remoto do Tivoli Workload Scheduler está instalado.

Exemplos

Neste exemplo, um aplicativo envia o tipo de evento customizado `BusProcCompleted` para um servidor do processador de eventos alternativo denominado `master3`. O evento é que o processamento do arquivo `calcweek` foi concluído.

```
sendevent -hostname master3 -port 4294 BusProcCompleted
GenericEventPlugIn TransacName=calcweek Workstation=ab5supp
```

O nome do arquivo e a estação de trabalho associada são os dois atributos do evento `BusProcCompleted` que foram especificados como atributos de evento acionadores em uma regra de evento associada.

showexec

Exibe os status dos jobs em execução. Este comando se aplica somente aoUNIX. Esse comando serve para os agentes padrão. Nos gerenciadores de domínio e nos agente tolerante a falhas utilize o comando **conman showjobs**.

Sintaxe

showexec [-V | -U | INFO]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

INFO Exibe o nome do arquivo de tarefa em vez do usuário, data e hora.

Resultados

A saída do comando está disponível em dois formatos: **padrão** e **INFO**.

Exemplos

Para exibir jobs em execução no formato **padrão**, execute o seguinte comando:

```
showexec
```

Para exibir tarefas em execução no formato **INFO**, execute o seguinte comando:

```
showexec INFO
```

Formato Padrão

CPU A estação de trabalho em que o job é executado.

Programar
O nome do fluxo de job no qual o job é executado.

Tarefa O nome do job.

Job# O número do job.

Usuário
O nome de usuário do job.

Data de Início
A data em que o job iniciou a execução.

Horário de Início
A hora em que o job iniciou a execução.

Tempo Decorrido (Estimado)
O tempo estimado, em minutos, no qual o job será executado.

Formato Info

CPU A estação de trabalho em que o job é executado.

Programar
O nome do fluxo de job no qual o job é executado.

Tarefa O nome do job.

Job# O número do job.

JCL O nome do arquivo do job.

shutdown

Para os processos do Tivoli Workload Scheduler e, opcionalmente, também para o WebSphere Application Server. Aplica-se apenas a estações de trabalho do Windows. Você deve ter acesso ***shutdown*** à estação de trabalho.

Sintaxe

shutdown [-V | -U] [-appsrv]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

-appsrv
Também para o WebSphere Application Server.

Comentários

Certifique-se de que o *TWS_user* que você está utilizando pertença ao grupo Administradores definido na estação de trabalho do Windows.

Exemplos

Para exibir o nome e a versão do comando, execute o seguinte comando:

```
shutdown -V
```

Para parar os processos do Tivoli Workload Scheduler e oWebSphere Application Server, execute o seguinte comando:

```
shutdown -appsrv
```

ShutDownLwa

Pare o agent. Não é necessário nenhum acesso específico à estação de trabalho. Execute este comando localmente no agent que deseja parar.

Sintaxe

ShutDownLwa

Argumentos

Nenhum argumento é necessário.

Exemplos

Para parar o agent, execute o seguinte comando:

```
ShutDownLwa
```

StartUp

Inicia **netman**, o processo de gerenciamento de redes do Tivoli Workload Scheduler.

Você deve ter acesso ***start*** para a estação de trabalho.

Sintaxe

StartUp [**-V** | **-U**]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

Comentários

No Windows, o serviço**netman** é iniciado automaticamente quando um computador é reiniciado. **StartUp** pode ser utilizado para reiniciar o serviço se ele for parado por algum motivo.

No UNIX, o comando **StartUp** pode ser executado automaticamente, sendo chamado a partir do arquivo `/etc/inittab`, de forma que a infra-estrutura do WebSphere Application Server e **netman** sejam iniciados toda vez que um computador for reinicializado. O comando **StartUp** poderá ser utilizado para reiniciar o**netman** se ele for parado por algum motivo.

O restante da árvore de processo pode ser reiniciado com os comandos

```
conman start
conman startmon
```

. Consulte conman “start” na página 437 para obter informações adicionais.

Exemplos

Para exibir o nome e a versão do comando, execute o seguinte comando:

```
StartUp -V
```

Para iniciar o processo **netman**, execute o seguinte comando:

```
StartUp
```

StartUpLwa

Inicia oagent.

Não é necessário nenhum acesso específico à estação de trabalho. Execute este comando localmente no agent que deseja iniciar.

Sintaxe

StartUpLwa

Argumentos

Nenhum argumento é necessário.

Exemplos

Para iniciar oagent, execute o seguinte comando:

```
StartUpLwa
```

tws_inst_pull_info

Esse é um script que produz informações sobre seu ambiente do Tivoli Workload Scheduler e sua estação de trabalho local e pode fazer uma captura instantânea dos dados do DB2 e do WebSphere Application Server no gerenciador de domínio principal, salvando-os como um pacote datado.

Ele também pode gerar um relatório contendo não apenas os resultados da captura instantânea, mas também vários parâmetros de configuração e de ambiente. A ferramenta é útil para descrever um problema ao Suporte de Software IBM. Para melhores resultados, deve ser executada assim que o problema for descoberto.

Comentários

Consulte o *Guia de Resolução de Problemas do IBM Tivoli Workload Scheduler* para obter informações adicionais sobre este comando.

version

Exibe informações sobre a liberação atual do Tivoli Workload Scheduler instalado no sistema. Este comando se aplica somente aoUNIX. As informações são extraídas de um arquivo de versão.

Sintaxe

version -V | -u | -h

version [**-a**] [**-f** *vfile*] [*file* [...]]

Argumentos

-V Exibe a exibição do comando e sai.

-u Exibe informações de utilização do comando e sai.

-h Exibe informações de auxílio do comando e sai.

-a Exibe informações sobre todos os arquivos de produtos. O padrão é exibir informações apenas sobre os arquivos especificados.

-f *vfile* Especifica o caminho e o nome do arquivo de versão se for diferente da configuração padrão. O padrão é um arquivo chamado**version.info** no diretório de job atual.

file Especifica os nomes de arquivos de produtos, separados por espaços, para os quais as informações sobre a versão são exibidas. O padrão é não exibir informações sobre os arquivos, ou, se **-a** for utilizado, exibirá todas as informações sobre os arquivos.

Resultados

O cabeçalho da saída contém o nome do produto, a versão, o sistema operacional, o nível de correção e a data de instalação. O restante da tela lista informações sobre o arquivo ou arquivos especificados. Os arquivos são listados no seguinte formato:

Arquivo
O nome do arquivo.

Revisão
O número de revisão do arquivo.

Correção
O nível de correção do arquivo, se houver.

Tamanho (bytes)
O tamanho do arquivo em bytes.

Checksum
O checksum do arquivo. Checksum é calculado utilizando o comando do UNIX **sum**. No AIX, **sum** é utilizado com o argumento **-o**.

Comentários

As informações do arquivo do Tivoli Workload Scheduler estão contidas no arquivo `version.info`. Esse arquivo é colocado no diretório *TWS_home*`/version` durante a instalação. O arquivo `version.info` está em um formato específico e não alterado.

É possível mover o arquivo `version.info` para outro diretório. No entanto, deverá incluir, em seguida, o argumento **-f** para localizar o arquivo.

Exemplos

Para exibir informações sobre a liberação do Tivoli Workload Scheduler instalado, execute o seguinte comando:

```
./version
```

Uma saída de amostra desse comando é:

```
IBM Tivoli Workload Scheduler/VERSÃO 9.21 (C) Copyright IBM Corp 1998, 2013

IBM Tivoli Workload Scheduler 9.2 UNIX
```

Para exibir informações sobre todos os arquivos, execute o seguinte comando:

```
version/version -a -f version/version.info
```

Para exibir informações sobre o arquivo `customize`, execute o seguinte comando:

```
cd version
./version customize
```

Para exibir informações sobre o arquivo `customize`, quando`version.info` estiver em `/apps/maestro`, execute o comando a seguir:

```
cd version
./version -f /apps/maestro/version.info customize
```

Comandos não suportados

Os seguintes comandos utilitários não suportados fornecem funções no Windows que são semelhantes aos comandos UNIX **ps** e **kill**. Eles poderão ser utilizados se utilitários semelhantes do Windows não estiverem disponíveis.

Sintaxe

listproc

killproc *pid*

Comentários

listproc
Exibe uma listagem tabular de processos no sistema.

killproc
Elimina o processo com o ID de processo *pid*.

Nota: Quando executado pelo Administrador, **killproc** é capaz de eliminar processos de sistema.

Capítulo 14. Usando os comandos do utilitário no ambiente dinâmico

Este capítulo descreve os comandos utilitários do Tivoli Workload Scheduler para o ambiente dinâmico. Enquanto alguns comandos executam no gerenciador de domínio principal ou em um gerenciador de domínio dinâmico, outros executam no agentes. Eles estão instalados no caminho*TWA_home*/TDWB/bin e executam com os sistemas operacionais UNIX e Windows. Utilize comandos utilitários a partir do prompt de comandos do sistema operacional, exceto o utilitário **jobprop** que é possível usar somente em uma definição de tarefa, conforme descrito em "Transmitindo variáveis configuradas usando o **jobprop** de uma tarefa para outra na instância do mesmo fluxo de tarefas" na página 484.

O Tabela 80 contém a lista dos comandos do utilitário e, para cada comando, sua descrição e o tipo de estação de trabalho na qual é possível executá-los.

Tabela 80. Lista de Comandos Utilitários para Estações de Trabalho Dinâmicas

Comando	Descrição	Tipo de estação de trabalho
exportserverdata	Efetua o download da lista de instâncias do broker de carga de trabalho dinâmica a partir do banco de dados Tivoli Workload Scheduler e muda um número da porta ou um nome do host.	gerenciador de domínio principal ou gerenciador de domínio dinâmico
importserverdata	Efetua o upload da lista de instâncias do broker de carga de trabalho dinâmica para o banco de dados Tivoli Workload Scheduler após editar o arquivo temporário para mudar um número da porta ou um nome do host.	gerenciador de domínio principal ou gerenciador de domínio dinâmico
jobprop	Configura os valores de variáveis em uma tarefa que você pode passar para a tarefa sucessiva na instância do mesmo fluxo de tarefas. Para obter mais informações sobre como utilizar esse utilitário em uma definição de tarefa, consulte "Transmitindo variáveis configuradas usando o **jobprop** de uma tarefa para outra na instância do mesmo fluxo de tarefas" na página 484. Ele é instalado no diretório <TWS_INSTALLATION_DIR>/TWS/bin e é executado em sistemas operacionais UNIX e Windows.	agentes
movehistorydata	Move os dados presentes no banco de dados Tivoli Workload Scheduler para as tabelas de archive	gerenciador de domínio principal ou gerenciador de domínio dinâmico
param	Cria, exibe e exclui variáveis e senhas do usuário em agentes dinâmicos.	agentes
resource	Cria, modifica, associa, consulta ou configura recursos online ou offline.	gerenciador de domínio principal, gerenciador de domínio dinâmico ou agentes
sendevent	Envia os eventos genéricos ao servidor do processador de eventos atualmente ativo.	gerenciador de domínio dinâmicos e agentes
twstrace	Modifica no tempo de execução as configurações para rastreio no agentes	agentes

Nota: Para remover arquivos de logs da tarefa para estações de trabalho de agentes dinâmicos, configure o valor da propriedade MaxAge no JobManager.ini. Para obter detalhes adicionais, consulte os *manuais do IBM Tivoli Workload Scheduler: Guia de administração - Configurando o agente - Configurando propriedades de ativadores comuns [Ativadores]*.

Arquivo de Configuração da Linha de Comandos

O arquivo CLIConfig.properties contém informações de configuração que são usadas ao digitar comandos. Por padrão, os argumentos requeridos ao digitar comandos são recuperados deste arquivo, a menos que sejam explicitamente especificados na sintaxe do comando.

© Copyright IBM Corp. 1999, 2014

O arquivo `CLIConfig.properties` é criado no momento da instalação e está localizado no gerenciador de domínio principal no seguinte caminho:
TWA_home`/TDWB/config`

O arquivo `CLIConfig.properties` contém os seguintes conjuntos de parâmetros:

Propriedades padrão do Dynamic Workload Broker

ITDWBServerHost
Especifica o endereço IP de Dynamic Workload Broker.

ITDWBServerPort
Especifica o número da porta Dynamic Workload Broker. O valor padrão é **9550**.

ITDWBServerSecurePort
Especifica o número da porta Dynamic Workload Broker quando a segurança é ativada. O valor padrão é **9551**.

use_secure_connection
Especifica se a conexão segura deve ser usada. O padrão é **false**.

Caminho e Nome de Arquivo KeyStore e trustStore

keyStore
Especifica o nome e o caminho de um arquivo KeyStore contendo chaves privadas. Um arquivo keyStore contém chaves públicas e chaves privadas. As chaves públicas são armazenadas como certificados do assinante, enquanto que as chaves privadas são armazenadas nos certificados pessoais. O valor padrão é /Certs/TDWBClientKeyFile.jks.

trustStore
Especifica o nome e o caminho do arquivo trustStore contendo chaves públicas. Um arquivo trustStore é um arquivo de banco de dados de chave que contém chaves públicas. A chave pública é armazenada como um certificado do assinante. As chaves são utilizadas para várias finalidades, incluindo autenticação e integridade dos dados. O valor padrão é /Certs/ TDWBClientTrustFile.jks.

As senhas para arquivos keyStore e trustStore

keyStorepwd
Especifica a senha para o arquivo KeyStore.

trustStorepwd
Especifica a senha para o arquivo trustStore.

Os tipos de arquivos para arquivos keyStore e trustStore

keyStoreType
Especifica o tipo de arquivo para o arquivo keyStore. O valor padrão é JKS.

trustStoreType
Especifica o tipo de arquivo para o arquivo trustStore. O valor padrão é JKS.

O ID do usuário e a senha padrão para Dynamic Workload Broker

tdwb_user
Especifica o nome de usuário para um usuário autorizado a executar operações no Dynamic Workload Broker quando a

segurança estiver ativada. O valor padrão é **ibmschedcli**. Essa senha deve ser definida previamente no IBM WebSphere. Para obter informações adicionais sobre as considerações de segurança, consulte o *Tivoli Workload Scheduler: Administration Guide*, *SC23-9113*.

tdwb_pwd
Especifica a senha para um usuário autorizado a executar operações no Dynamic Workload Broker quando a segurança estiver ativada. Essa senha deve ser definida previamente no IBM WebSphere. Para obter informações adicionais sobre segurança, consulte o *Tivoli Workload Scheduler: Administration Guide*.

O nível de detalhe para informações de rastreio e de log de linha de comandos

logger.Level
Especifica o nível de detalhe para o rastreio de linha de comandos e os arquivos de log. O rastreio de linha de comandos e os arquivos de log são criados no seguinte local:

arquivo de log
TWA_home/TDWB/logs/Msg_cli.log.log

arquivo de rastreio
TWA_home/TDWB/logs/Trace_cli.log

O valor padrão é INFO.

logger.consoleLevel
Especifica o nível de detalhe para as informações de rastreio e log a serem retornadas para a saída padrão. O valor padrão é FINE. Os valores suportados para os parâmetros **consoleLevel** e **loggerLevel** são os seguintes:

ALL Indica que todas as mensagens são registradas.

SEVERE
Indica que as mensagens de erro graves são registradas.

WARNING
Indica que as mensagens de aviso são registradas.

INFO Indica que as mensagens informativas são registradas.

CONFIG
Indica que as mensagens de configuração estáticas são registradas.

FINE Indica que as informações de rastreio são registradas.

FINER
Indica que as informações de rastreio detalhadas são registradas.

FINEST
Indica que as informações de rastreio altamente detalhadas são registradas.

OFF Indica que o registro está desativado.

logger.limit
Especifica o tamanho máximo de um arquivo de log em bytes. O valor padrão é 400000. Quando o tamanho máximo é alcançado, um novo arquivo é criado, até que o número máximo de arquivos

seja alcançado. Quando todos os arquivos atingirem o tamanho máximo e o número máximo de arquivos for excedido, o arquivo mais antigo será regravado.

logger.count
Especifica o número máximo de arquivos de log. O valor padrão é 6. Quando o tamanho máximo é alcançado, um novo arquivo é criado, até que o número máximo de arquivos seja alcançado. Quando todos os arquivos atingirem o tamanho máximo e o número máximo de arquivos for excedido, o arquivo mais antigo será regravado. Quando um novo arquivo é criado, o sufixo 0 é anexado após a extensão do arquivo. O arquivo com o sufixo 0 é sempre o arquivo atual. Qualquer arquivo mais antigo será renumerado de forma adequada.

java.util.logging.FileHandler.pattern
Especifica que as informações de rastreio para Java Virtual Machine estão registradas em log no arquivo Trace_cli.log. O valor padrão é INFO.

java.util.logging.FileHandler.limit
Especifica o tamanho máximo de um arquivo de rastreio em bytes. O valor padrão é 400000. Quando o tamanho máximo é alcançado, um novo arquivo é criado, até que o número máximo de arquivos seja alcançado. Quando todos os arquivos atingirem o tamanho máximo e o número máximo de arquivos for excedido, o arquivo mais antigo será regravado.

java.util.logging.FileHandler.count
Especifica o número máximo de arquivos de rastreio. O valor padrão é 6. Quando o tamanho máximo é alcançado, um novo arquivo é criado, até que o número máximo de arquivos seja alcançado. Quando todos os arquivos atingirem o tamanho máximo e o número máximo de arquivos for excedido, o arquivo mais antigo será regravado. Quando um novo arquivo é criado, o sufixo 0 é anexado após a extensão do arquivo. O arquivo com o sufixo 0 é sempre o arquivo atual. Qualquer arquivo mais antigo será renumerado de forma adequada.

java.util.logging.FileHandler.formatter
Especifica o formatador a ser utilizado para o arquivo Trace_cli.log. O valor padrão é com.ibm.logging.icl.jsr47.CBEFormatter.

Configuração Comum DAO
Esta seção define as configurações RDBMS para o comando **movehistorydata**. Esses comandos usam os RDBMS instalados no Dynamic Workload Broker Esses parâmetros são valorizados no tempo de instalação e não devem ser modificados, exceto para `com.ibm.tdwb.dao.rdbms.useSSLConnections` conforme observado abaixo.

com.ibm.tdwb.dao.rdbms.rdbmsName
Especifica o nome RDBMS.

com.ibm.tdwb.dao.rdbms.useDataSource
Especifica a origem de dados a ser utilizada.

com.ibm.tdwb.dao.rdbms.jdbcPath
Especifica o caminho ao driver JDBC.

com.ibm.tdwb.dao.rdbms.jdbcDriver
Especifica o driver JDBC.

com.ibm.tdwb.dao.rdbms.userName
Especifica o nome do usuário RDBMS.

com.ibm.tdwb.dao.rdbms.password
Especifica a senha do usuário RDBMS.

com.ibm.tdwb.dao.rdbms.useSSLConnections
Especifica que o acesso para o banco de dados Tivoli Workload Scheduler DB2 por alguns dos comandos CLI é via SSL. É configurado para `FALSE` por padrão. É necessário configurar para `TRUE`, se o banco de dados for DB2 e você usar segurança FIPS, para que os seguintes comandos funcionem:

- **exportserverdata**
- **importserverdata**
- **movehistorydata**

exportserverdata

Utilize o comando **exportserverdata** para fazer download da lista de instância do dynamic workload brokers a partir do banco de dados do Tivoli Workload Scheduler e altere um número da porta ou nome do host.

Sintaxe

exportserverdata ?

exportserverdata -dbUsr *db_user_name* **-dbPwd** *db_user_password* **-exportFile** *filename*

Descrição

Este comando extrai uma lista de URIs (Uniform Resource Identifier) de todos os instância do dynamic workload brokers do banco de dados Tivoli Workload Scheduler e os copia para um arquivo temporário de maneira que, se o nome do host ou número de porta de qualquer instância relacionada for alterado, o administrador pode registrar essas informações no arquivo e colocá-las de volta no banco de dados com o comando `importserverdata`. Por padrão, a lista de URIs é salva no arquivo server.properties, localizado no diretório atual.

Essa ação é necessária porque a lista de instância do dynamic workload brokers deve ser mantida atualizada sempre, pois os agentes do Resource Advisor conectam periodicamente à instância ativa para enviar seus dados sobre os recursos descobertos em cada computador. Eles são capazes de alternar automaticamente entre as instâncias dessa lista e localizar uma ativa para copiar esses dados em seu Resource Repository. Como o gerenciador de domínio principal e cada backup principal estão instalados com um instância do dynamic workload broker, o instância do dynamic workload broker ativo executa no gerenciador de domínio principal, enquanto uma instância inativa reside em cada backup principal.

A URI que aponta para cada instância do dynamic workload broker é a seguinte:
`https://`*`hostname`*`:`*`port_number`*`/JobManagerRESTWeb/JobScheduler`

Você pode alterar somente o nome do host e o número da porta.

Importante: A lista é ordenada. É possível alterar a ordem das instâncias conforme elas aparecem nessa lista, e os agentes seguirão essa ordem. Se você tiver vários principais de backup e decidir seguir uma ordem alternada específica quando um principal falhar, poderá instruir os agentes para alternarem para a instância correta usando essa lista ordenada, aumentando a velocidade do tempo de transição.

Se seu banco de dados Tivoli Workload Scheduler for DB2 e você usar a segurança FIPS, para executar esse comando com êxito é necessário ter a opção **com.ibm.tdwb.dao.rdbms.useSSLConnections** configurada para `TRUE` no arquivo `CLIConfig.properties`.

Opções

? Exibe informação de ajuda.

-dbUsr *db_user_name*
O nome de usuário necessário para acessar o servidor de banco de dados Tivoli Workload Scheduler.

-dbPwd *db_user_password*
A senha de usuário necessária para acessar o servidor de banco de dados Tivoli Workload Scheduler.

-exportFile *filename*
O nome do arquivo temporário onde as URIs extraídas do banco de dados são copiadas para edição. Esse arquivo de texto é criado quando você executa o comando e é possível abri-lo com qualquer editor para alterar o nome do host e o número da porta. Se você não especificar um caminho, o arquivo será criado no mesmo diretório em que o comando está localizado, ou seja:

```
<TWA_home>/TDWB/bin
```

Se você especificar um caminho diferente, certifique-se de que o caminho existe antes de executar o comando.

Exemplo

Para fazer download da lista atual de instância do dynamic workload brokers (ativo e backup) e copiá-los em um arquivo denominado `c:\myservers\uris160709`, execute:

```
exportserverdata -dbUsr twsadm
-dbPwd fprefect -exportFile c:\myservers\uris160709
```

O comando retorna o arquivo `uris160709`, que é semelhante a este:

```
https://accrec015:42127/JobManagerRESTWeb/JobScheduler
https://prodop099:52529/JobManagerRESTWeb/JobScheduler
https://prodop111:31116/JobManagerRESTWeb/JobScheduler
```

`prodop099` é o instância do dynamic workload broker ativo porque está hospedado pelo gerenciador de domínio principal ativo atualmente, ao passo que `accrec015` e `prodop111` estão inativos porque estão hospedados por backup principal.

É possível editar esse arquivo para aplicar as alterações antes de usar o comando `importserverdata` para fazer upload das URIs de volta ao banco de dados.

Consulte Também

"importserverdata" na página 539

importserverdata

Utilize o comando **importserverdata** para fazer upload da lista de instância do dynamic workload brokers para o banco de dados do Tivoli Workload Scheduler após editar o arquivo temporário para alterar um número de porta ou um nome de host.

Sintaxe

importserverdata ?

importserverdata -dbUsr *db_user_name* **-dbPwd** *db_user_password* **-importFile** *filename*

Descrição

Este comando retorna a lista de instância do dynamic workload brokers no banco de dados do Tivoli Workload Scheduler a partir de um arquivo temporário onde eles foram previamente transferidos por download com o comando `exportserverdata`.

Utilize os comandos `exportserverdata` e `importserverdata` se precisar registrar algumas mudanças de nome do host ou número da porta nas URIs das instâncias. Isso é necessário para manter a lista de instância do dynamic workload brokers atualizada sempre, pois os agentes do Resource Advisor conectam periodicamente à instância ativa para enviar seus dados sobre os recursos descobertos em cada computador. Eles são capazes de alternar automaticamente entre as instâncias dessa lista e localizar uma ativa para copiar esses dados em seu Resource Repository. Como o gerenciador de domínio principal e cada backup principal estão instalados com um instância do dynamic workload broker, o instância do dynamic workload broker ativo executa no gerenciador de domínio principal, enquanto uma instância inativa reside em cada backup principal.

Importante: A lista é ordenada. É possível alterar a ordem das instâncias conforme elas aparecem nessa lista, e os agentes seguirão essa ordem. Se você tiver vários principais de backup e decidir seguir uma ordem alternada específica quando um principal falhar, poderá instruir os agentes para alternarem para a instância correta usando essa lista ordenada, aumentando a velocidade do tempo de transição.

Se seu banco de dados Tivoli Workload Scheduler for DB2 e você usar a segurança FIPS, para executar esse comando com êxito é necessário ter a opção **com.ibm.tdwb.dao.rdbms.useSSLConnections** configurada para `TRUE` no arquivo `CLIConfig.properties`.

Opções

? Exibe informação de ajuda.

-dbUsr *db_user_name*
O nome de usuário necessário para acessar o servidor de banco de dados Tivoli Workload Scheduler.

-dbPwd *db_user_password*
A senha de usuário necessária para acessar o servidor de banco de dados Tivoli Workload Scheduler.

-importFile f*filename*
O nome do arquivo temporário que você especificou com a palavra-chave `-exportFile` no comando `exportserverdata`.

Exemplo

Para fazer upload da lista editada de URLs do instância do dynamic workload broker a partir do arquivo `c:\myservers\uris160709` para o banco de dados Tivoli Workload Scheduler, execute:

```
importserverdata -dbUsr twsadm
-dbPwd fprefect -importFile c:\myservers\uris160709
```

Consulte Também

"exportserverdata" na página 537

jobprop

Use o comando **jobprop** em uma definição de tarefa para configurar variáveis localmente para uma tarefa em agentes dinâmicos e no Agentes do Tivoli Workload Scheduler for z/OS.

É possível utilizar este comando em uma tarefa nativa ou executável para configurar o valor da variável que se pode passar em uma tarefa sucessiva no mesmo fluxo de tarefas. Os valores são configurados no tempo de execução.

Sintaxe

jobprop *variable value*

Argumentos

variável
O nome da variável.

valor O valor para *variable*.

Comentários

Os nomes fazem distinção entre maiúsculas e minúsculas.

Exemplos

Em sistemas operacionais UNIX, o utilitário **jobprop** configura as seguintes variáveis na tarefa executável `NC125133#JOBA`:

- *VAR1* variável configurada para o valor `value1`.
- *VAR2* variável configurada para o valor `value2`.
- *VAR3* variável configurada para o valor `value3`.
- *VAR4* variável configurada para o valor `value4`.

```
NC125133#JOBA
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl" xmlns:
jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle">
  <jsdl:application name="executable">
    <jsdle:executable interactive="false">
    <jsdle:script>#!/bin/sh
. /home/ITAuser/TWA/TWS/tws_env.sh
jobprop VAR1 value1
jobprop VAR2 value2
jobprop VAR3 value3
jobprop VAR4 value4
```

```
</jsdle:script>
    </jsdle:executable>
  </jsdl:application>
</jsdl:jobDefinition>
 DESCRIPTION "Sample Job Definition"
 RCCONDSUCC "RC>=0"
 RECOVERY STOP
```

Em sistemas operacionais Windows, o utilitário **jobprop** configura as seguintes variáveis na tarefa executável WIN1#JOB1:

- *var1* variável configurada para o valor value1.
- *var2* variável configurada para o valor value2.
- *var3* variável configurada para o valor value3.
- *var4* variável configurada para o valor value4.

```
WIN1#JOB1
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl" xmlns:
jsdle="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdle">
  <jsdl:application name="executable">
    <jsdle:executable interactive="false">
    <jsdle:script>
call C:\Progra~1\IBM\TWA\TWS\tws_env.cmd
jobprop var1 value1
jobprop var2 value2
jobprop var3 value3
jobprop var4 value4
</jsdle:script>
    </jsdle:executable>
  </jsdl:application>
</jsdl:jobDefinition>
 DESCRIPTION "Sample Job Definition"
 RCCONDSUCC "RC>=0"
 RECOVERY STOP
```

Nota: Antes de executar o utilitário **jobprop** na definição de tarefa, assegure-se de executar o comando **tws_env.cmd** com a sintaxe correta `call <TWS_INST_DIR>\TWS\tws_env.cmd`, em que <*TWS_INST_DIR*> é o diretório de instalação do Tivoli Workload Scheduler.

movehistorydata

Use o **movehistorydata** quando o acesso ao banco de dados se tornar muito lento. Esse comando move os dados presentes no Repositório de Tarefas para as tabelas de archive

O acesso lento ao banco de dados pode ser devido a um grande número de registros presentes no banco de dados, por exemplo, quando envios de tarefa em massa são executados.

Quando você executa esse comando, as tarefas são movidas para as seguintes tabelas no banco de dados:

JOA_JOB_ARCHIVES
Contém as instâncias de tarefas arquivadas

JRA_JOB_RESOURCE_ARCHIVES
Contém as informações de recursos relacionadas às tarefas

MEA_METRIC_ARCHIVES
Contém as métricas coletadas para as tarefas

Para obter informações adicionais sobre as tabelas de históricos, consulte o *Tivoli Workload Scheduler: Administration Guide*, *SC23-9113*.

Nota: Dependendo do número de tarefas e acessos ao banco de dados, uma operação de limpeza pode causar alguns picos na memória ou no uso da CPU.

Se seu banco de dados Tivoli Workload Scheduler for DB2 e você usar a segurança FIPS, para executar esse comando com êxito é necessário ter a opção **com.ibm.tdwb.dao.rdbms.useSSLConnections** configurada para `TRUE` no arquivo `CLIConfig.properties`.

Sintaxe

movehistorydata ?

movehistorydata -dbUsr *db_user_name***-dbPwd** *db_user_password* [**-successfulJobsMaxAge** *successfulJobsMaxAge* [**-notSuccessfulJobsMaxAge** *notSuccessfulJobsMaxAge*][**-archivedJobsMaxAge** *archivedJobsMaxAge*]]

Descrição

Este comando desempenha uma operação de limpeza na banco de dados do Repositório de Tarefas. Com base nos valores especificados, as informações sobre tarefas enviadas são movidas para o banco de dados de arquivamento e as informações nesse banco de dados são excluídas.

Utilize esse comando para substituir temporariamente as configurações definidas no arquivo **JobDispatcherConfig.properties**, quando eventos inesperados requerem uma limpeza imediata do banco de dados. As configurações no arquivo **JobDispatcherConfig.properties** permanecem inalteradas. Para obter mais informações sobre o arquivo **JobDispatcherConfig.properties**, consulte *Tivoli Workload Scheduler: Administration Guide*.

Opções

? Exibe informação de ajuda.

-dbUsr *db_user_name*
Especifica o nome de um usuário autorizado a executar operações no servidor de banco de dados.

-dbPwd *db_user_password*
Especifica a senha de um usuário autorizado a executar operações no servidor de banco de dados.

-successfulJobsMaxAge *successfulJobsMaxAge*
Especifica quantas horas as tarefas concluídas com êxito ou canceladas deverão ser mantidas no banco de dados do Repositório de Tarefas antes de serem arquivadas. O valor padrão é 240 horas, ou seja, dez dias.

-notSuccessfulJobsMaxAge *notSuccessfulJobsMaxAge*
Especifica quantas horas as tarefas concluídas sem êxito ou com um status desconhecido deverão ser mantidas no banco de dados do Repositório de Tarefas antes de serem arquivadas. O valor padrão é 720 horas, ou seja, 30 dias.

-archivedJobsMaxAge *archivedJobsMaxAge*
Especifica quantas horas as tarefas devem ser mantidas no banco de dados de arquivamento antes de serem arquivadas. O valor padrão é 720 horas, ou seja, 30 dias.

Valores de Retorno

O comando **movehistorydata** retorna um dos seguintes valores:
0 Indica que **movehistorydata** foi concluído com êxito.

< > 0 Indica que **movehistorydata** falhou.

Exemplos

1. Para mover todas as tarefas concluídas com êxito para o banco de dados de arquivamento nas últimas 40 horas, digite o seguinte comando:

   ```
   movehistorydata -dbUsr halmst -dbPwd dgordon -successfulJobsMaxAge 40
   ```

2. Para mover todas as tarefas com status suportados para o banco de dados de arquivamento e remover todas as tarefas anteriores a 700 horas do banco de dados de arquivamento, digite o seguinte comando:

   ```
   movehistorydata -dbUsr halmst -dbPwd dgordon -successfulJobsMaxAge 0
                   -notSuccessfulJobsMaxAge 0 -archivedJobsMaxAge 700
   ```

param

Use o comando **param** para definir e gerenciar senhas de usuário e variáveis localmente em agentes dinâmicos e no Agentes do Tivoli Workload Scheduler for z/OS.

É possível usar esse comando em tipos de tarefas com opções avançadas. Os valores são resolvidos no tempo de envio no agente no qual a tarefa é enviada.

Nota: No Windows 2012, o comando não é suportado no Windows PowerShell.

Autorização

Para criar, excluir ou exibir variáveis ou senhas, você deve ter direitos de Administrador ou de usuário raiz na estação de trabalho que executa o agente ou direitos de *TWS_user* no agente.

Sintaxe

param -u | -V |
{-c | -ec} [*file.section.*|*file..*|*section.*]
variable [*value*] |
[*file.section.*|*file..*|*section.*] *variable* |
{-d | -fd} [*file.section.*|*file..*|*section.*]
variable

Argumentos

-u Exibe informações de utilização do comando e sai.

-V Exibe a versão do comando e sai.

-c | -ec
Cria variável ou senha *variable* e define seu valor *value*. A variável ou senha é colocada em um namespace *file* que você pode organizar em uma ou mais seções nomeadas *section*.

Se você não fornecer um nome de arquivo *file*, a variável ou senha será colocada no arquivo padrão `jm_variables` no caminho*agent_installation_path*`\TWA\TWS\ITA\cpa\config\jm_variables_files` (`/TWA/TWS/ITA/cpa/config/jm_variables_files`) no agente dinâmico.

Se você não fornecer um nome de seção *section*, a variável ou senha será colocada no corpo principal do arquivo.

Importante: Se estiver definindo uma senha, você deve especificar uma seção nomeada `password` para *variable*. Isso especifica que *variable* é uma senha.

Se estiver criando uma variável, *variable* é o nome da variável e *value* é seu valor. Se estiver criando uma senha, *variable* é o nome de usuário e *value* é sua senha. Se não inserir *value* dentro dos argumentos, o comando solicitará interativamente a inserção de um valor.

O argumento **-c** cria a variável em formulário limpo. O argumento **-ec** cria a variável em formulário criptografado. As senhas são criptografadas por padrão também se você usar **-c**.

-d | -fd
Exclui (**-d**) ou força a exclusão (**-fd**) de um arquivo, seção ou variável (senha). É possível usar os seguintes curingas:

***** Substitui um ou mais caracteres alfanuméricos.

? Substitui um caractere alfanumérico.

Com **-d** o comando pede confirmação antes de excluir. Com **-fd** ele exclui sem pedir confirmação.

Ao excluir todas as variáveis em uma seção, a seção é removida do arquivo. Ao excluir todas as seções e todas as variáveis de um arquivo, o arquivo é removido.

file O nome do arquivo usado como um namespace para *variable*. Se você não especificar *file*, o comando usará o arquivo padrão `jm_variables` no caminho*agent_installation_path*`\TWA\TWS\ITA\cpa\config\jm_variables_files` (`/TWA/TWS/ITA/cpa/config/jm_variables_files`).

Todos os namespaces de variável vão no caminho*agent_installation_path*`\TWA\TWS\ITA\cpa\config\jm_variables_files` (`/TWA/TWS/ITA/cpa/config/jm_variables_files`).

section O nome da seção dentro de *file* em que *variable* é definida. Quando*variable* é usada para uma senha, ela deve ser colocada em uma seção nomeada `password`. Nenhum nome de seção é requerido para armazenar variáveis.

valor O valor para *variable*.

variável
Pode ser um nome de variável ou uma identificação de usuário. Se for usada para identificação, ela deve ser colocada em uma seção nomeada `password` dentro do arquivo do namespace.

Comentários

Para exibir uma variável ou senha, um arquivo de namespace, ou uma seção, usará o comando como a seguir:

```
param [file.section.|file..|section.] variable
```

em que é possível usar os curingas * e ? descritos para o comando de exclusão.

Os arquivos de namespace, incluindo o`jm_variables` padrão, não possuem extensão.

Os nomes de variável fazem distinção entre maiúsculas e minúsculas.

Em sistemas IBM i, se você usar as shells QP2TERM e QSH, as senhas são tornadas visíveis durante o processo de criação com `param` e são exibidas claramente nos logs da shell. Para garantir a ofuscação de uma senha, você precisa usar a shell AIXTERM ou XTERM.

Exemplos

O comando:

```
param -c
compassets.hardware.platform1 unix
```

define a variável `platform1` com o valor `unix` na seção `hardware` do arquivo novo ou existente nomeado`compassets`. O valor não é criptografado.

O comando:

```
param -c
compassets..platform1 unix
```

define a variável `platform1` com o valor `unix` no arquivo novo ou existente nomeado`compassets`. O valor não é criptografado.

O comando:

```
param -ec
hardware.platform1 unix
```

define a variável `platform1` com o valor `unix` na seção `hardware` no arquivo padrão *agent_installation_path*`\TWA\TWS\ITA\cpa\config\jm_variables_files\jm_variables`. O valor é criptografado.

O comando:

```
param -c
compassets.password.jladams san07rew
```

define a variável `jladams` com o valor `san07rew` na seção `password` do arquivo novo ou existente nomeado`compassets`. Como `jladams` é definido na seção `password`, ele é interpretado como um nome de usuário. O valor `san07rew` é criptografado por padrão, pois ele é interpretado como uma senha.

O comando:

```
param *.*.platform1
```

lista a variável `platform1` em todos os seus locais definidos. Isto é:

```
...\TWA\TWS\ITA\cpa\config\jm_variables_files\compassets.hardware.platform1=unix
...\TWA\TWS\ITA\cpa\config\jm_variables_files\compassets..platform1=unix
...\TWA\TWS\ITA\cpa\config\jm_variables_files\jm_variables.hardware.platform1=***
```

O comando:

```
param password.*adam*
```

lista todas as variáveis incluindo a sequência `adam` contida na seção `password` de todos os arquivos. Neste caso:

```
...\TWA\TWS\ITA\cpa\config\jm_variables_files\compassets.password.jladams********
```

O comando:

```
param -d compassets.password.jladams
```

exclui a variável `jladams`.

O comando:

```
param -d
compassets.password.*
```

exclui todas as variáveis localizadas na seção `password` e, portanto, remove essa seção do arquivo `compassets`.

O comando:

```
param -d compassets.*.*
```

exclui todo o conteúdo (variáveis e seções que contêm variáveis) localizadas no arquivo `compassets` e, portanto, remove o arquivo.

---

recurso

Use o comando **resource** para criar, modificar, associar, consultar ou configurar recursos online ou offline.

Ao configurar corretamente o arquivo **`CLIConfig.properties`** no agente, é possível executar esse comando também a partir de qualquer agente Tivoli Workload Scheduler conectado. Consulte "Utilizando o comando resource a partir de um agente" na página 554 para obter detalhes.

Sintaxe

resource ?

resource [**-usr** *user_name* **-pwd** *password*]
{
[**-create**{ **-logical** *name* **-type** *type*[**-quantity** *quantity*][**-offline**] |
-group *name*[**-offline**]}]
|
[**-delete**{**-logical** *name* |
-group *name* }]
|
[**-update**{**-computer** *name*{[**-setOnline** | **-setOffline**]} |
-logical *name*
[**-setName** *name*]
[**-setType** *type*]
[**-setQuantity** *quantity*]
[**-setOnline** | **-setOffline**]
[**-addComputer** *name* |
-addComputerByID *ID* |
-removeComputer *name* |
-removeComputerByID *ID*]
|
-group *name*
[**-setName** *name*]
[**-setOnline** | **-setOffline**]
[**-addComputer** *name* |
-addComputerByID *ID* |
-removeComputer *name* |
-removeComputerByID *ID* |
-addLogical *name* |
-removeLogical *name*]}]
|
[**-query**{**-computer** *name* [*-v*] |
-logical *name* [*-v*] |

-group *name* [*-v*]}
[**-configFile** *configuration_file*]
}

Descrição

Use esse comando para trabalhar com computadores, recursos lógicos e grupos de recursos. Em particular, é possível:

- Criar, atualizar, listar e excluir recursos lógicos ou grupos
- Criar recursos lógicos, associá-los aos computadores, definir grupos de recursos lógicos ou computadores e defini-los online ou offline
- Recuperar e atualizar propriedades do recurso usando a consulta e as opções de atualização.
- Descubra a lista de computadores associados a um recurso lógico executando uma consulta detalhada no recurso lógico.
- Alterar a associação entre computadores e recursos lógicos
- Configurar recursos online ou offline e consultar propriedades do computador

Opções

? Exibe informação de ajuda.

-usr *user_name*
Especifica o nome de um usuário autorizado a desempenhar operações na linha de comandos. Essa opção é necessária quando a segurança está ativada e o nome de usuário não está definido no arquivo de configuração `CLIConfig.properties` (com a palavra-chave `tdwb_user`).

-pwd *password*
Especifica a senha de um usuário autorizado a desempenhar operações na linha de comandos. Essa opção é necessária quando a segurança está ativada e a senha não está definida no arquivo de configuração `CLIConfig.properties` (com a palavra-chave `tdwb_pwd`).

-create -logical *name* **-type** *type*
Cria o recurso lógico com o nome e o tipo especificados. Também é possível configurar uma quantidade específica ou configurar o recurso offline usando parâmetros opcionais da seguinte maneira:

-create -logical *name* **-type** *type***-quantity** *quantity* **-offline**

-create -group *name*
Cria o grupo de recursos com o nome especificado. Também é possível configurá-lo offline usando o parâmetro opcional `-offline` da seguinte maneira:

-create -group *name* **-offline**

-delete -logical *name*
Exclui o recurso lógico com o nome especificado.

-delete -group *name*
Exclui o grupo de recursos com o nome especificado.

-update -computer *name*
Atualiza o sistema do computador com o nome especificado. É possível configurar o computador online ou offline da seguinte maneira:

-update -computer *name* **-setOnline**
Configura o computador especificado online.

-update -computer *name* **-setOffline**
Configura o computador especificado offline.

-update -logical *name*
Atualiza o recurso lógico especificado. É possível atualizar as propriedades e o status de um recurso das seguintes maneiras:

-update -logical *name* **-setName** *name*
Atualiza o nome do recurso lógico especificado.

-update -logical *name* **-setType** *type*
Atualiza o tipo do recurso lógico especificado.

-update -logical *name* **-setQuantity** *quantity*
Atualiza a quantidade do recurso lógico especificado.

-update -logical *name* **-setOnline**
Configura online o recurso lógico especificado.

-update -logical *name* **-setOffline**
Configura offline o recurso lógico especificado.

É possível alterar a associação entre um recurso lógico e um computador das seguintes maneiras:

-update -logical *name* **-addComputer** *name*
Associa o recurso lógico especificado ao computador com o nome especificado.

-update -logical *name* **-addComputerByID** *ID*
Associa o recurso lógico especificado ao computador com o ID especificado.

-update -logical *name* **-removeComputer** *name*
Remove a associação entre o recurso lógico especificado e o computador com o nome especificado.

-update -logical *name* **-removeComputerByID** *ID*
Remove a associação entre o recurso lógico especificado e o computador com o ID especificado.

-update -group *name*
Atualiza o grupo de recursos especificado. É possível atualizar as propriedades e o status de um grupo de recursos das seguintes maneiras:

-update -group *name* **-setName** *name*
Atualiza o nome do grupo de recursos especificado.

-update -group *name* **-setOnline**
Configura online o grupo de recursos especificado.

-update -group *name* **-setOffline**
Configura offline o grupo de recursos especificado.

É possível incluir recursos lógicos ou computadores em um grupo de recursos ou removê-los do grupo de recursos das seguintes maneiras:

-update -group *name* **-addLogical** *name*
Inclui o recurso lógico com o nome especificado no grupo de recursos.

-update -group *name* **-removeLogical** *name*
Remove o recurso lógico com o nome especificado do grupo de recursos.

-update -group *name* **-addComputer** *name*
Inclui o computador com o nome especificado no grupo de recursos.

-update -group *name* **-addComputerByID** *ID*
Inclui o computador com o ID especificado no grupo de recursos.

-update -group *name* **-removeComputer** *name*
Remove o computador com o nome especificado do grupo de recursos.

-update -group *name* **-removeComputerByID** *ID*
Remove o computador com o ID especificado do grupo de recursos.

-query -computer *name*
Recupera as seguintes propriedades do computador especificado:

- Per.
- ID do Computador
- Nome do sistema operacional
- Tipo de Sistema Operacional
- Versão do sistema operacional
- Status
- Status de disponibilidade

Recupera as seguintes propriedades adicionais, se você incluir a opção **-v**:

- Memória física
- Memória virtual
- Utilização de CPU
- Memória física livre
- Memória virtual livre
- Espaço de troca livre
- Memória física alocada
- Memória virtual alocada
- Espaço de troca alocado
- Número de processadores
- Número de processadores alocados
- Tipo de processador
- Velocidade do processador
- Fabricante
- Modelo
- Número de série
- Interfaces de rede
- Sistemas de arquivos

É possível usar o asterisco (*) como um caractere curinga das seguintes maneiras:

Como um parâmetro único
Você deve colocá-lo entre aspas duplas, por exemplo:

```
C:\IBM\TWA\TDWB\bin>resource –query –computer "*"
```

Esse comando retorna uma lista de todos os computadores existentes.

Para completar um nome do computador
Você deve colocar o nome inteiro entre aspas duplas, por exemplo:

```
C:\IBM\TWA\TDWB\bin> resource –query –computer "lab123*"
```

Esse comando retorna uma lista de todos os computadores existentes com um nome que inicia com `lab123`.

-query -logical *name*
Recupera o nome e o tipo do recurso lógico especificado. Recupera as seguintes propriedades adicionais, se você incluir a opção **-v**:

- Status
- Quantidade
- Alocação atual
- Lista de computadores

É possível usar o asterisco (*) como um caractere curinga das seguintes maneiras:

Como um parâmetro único
Você deve colocá-lo entre aspas duplas, por exemplo:

```
C:\IBM\TWA\TDWB\bin>resource –query –logical "*"
```

Esse comando retorna uma lista de todos os recursos lógicos existentes.

Para completar um nome do recurso
Você deve colocar o nome inteiro entre aspas duplas, por exemplo:

```
C:\IBM\TWA\TDWB\bin> resource –query –logical "myRes*"
```

Esse comando retorna uma lista de todos os recursos lógicos existentes com um nome que inicia com `myRes`.

-query -group *name*
Recupera o nome e o status do grupo de recursos especificado. Recupera a lista de computadores e de recursos lógicos contidos no grupo de recursos, se você usar a opção **–v**.

É possível usar o asterisco (*) como um caractere curinga das seguintes maneiras:

Como um parâmetro único
Você deve colocá-lo entre aspas duplas, por exemplo:

```
C:\IBM\TWA\TDWB\bin>resource –query –group "*"
```

Esse comando retorna uma lista de todos os grupos de recursos existentes.

Para completar um nome do grupo de recursos
Você deve colocar o nome inteiro entre aspas duplas, por exemplo:

```
C:\IBM\TWA\TDWB\bin> resource –query –group "myResGrou*"
```

Esse comando retorna uma lista de todos os grupos de recursos existentes com um nome que inicia com `myResGrou`.

-configFile *configuration_file*
Especifica o nome e o caminho de um arquivo de configuração customizado. Essa palavra-chave é opcional. Se ela não for especificada, o arquivo de configuração padrão será assumido. Para obter informações adicionais sobre o arquivo de configuração, consulte a seção sobre **CLIConfig.properties**.

Autorização

O nome de usuário e a senha para o comando são definidos no arquivo `CLIConfig.properties`. Para substituir as configurações definidas nesse arquivo, insira o nome de usuário e a senha ao digitar o comando. Para obter informações

adicionais sobre o arquivo CLIConfig.properties, consulte a seção sobre o**CLIConfig.properties**.

Valores de Retorno

O comando **resource** retorna um dos seguintes valores:

0 Indica que o comando foi concluído com êxito.

< > 0 Indica que o comando falhou.

Exemplos

- Para criar um recurso lógico denominadomyApplication, do tipo Aplicativos, digite o seguinte comando:

```
resource.bat -usr john -pwd
BXVFDCGS -create -logical myApplication
-type Applications
```

  A seguinte saída é exibida:

```
AWKCLI153I Logical resource "myApplication" created.
```

- Para atualizar a quantidade do recurso lógico denominadomyApplication, digite o seguinte comando:

```
resource.bat -update -logical myApplication -setQuantity 5
-usr john -pwd BXVFDCGS
```

  A seguinte saída é exibida:

```
AWKCLI165I Logical resource "myApplication" updated.
```

- Para incluir o relacionamento entre um recurso lógico e um computador, digite o seguinte comando:

```
resource.bat -update -logical myApplication -addComputer myComputer
-usr john -pwd BXVFDCGS
```

  A seguinte saída é exibida:

```
AWKCLI165I Logical resource "myApplication" updated.
```

- Para recuperar os detalhes de um recurso lógico denominadomyApplication, digite o seguinte comando:

```
resource.bat -usr john -pwd BXVFDCGS -query -logical myApplication –v
```

  A seguinte saída é exibida:

```
AWKCLI171I Calling the resource repository to perform a query on resources.

AWKCLI172I "1" logical resources were found for your query.
Details are as follows:

Resource Name:myApplication
Resource Type:Applications
Resource Status:Online
Resource Quantity:5
Resource Current Allocation:0
Computers List:
        Computer Name:myComputer
        Computer ID:D656470E8D76409F9F4FDEB9D764FF59
        Computer Status:Online
        Computer Availability Status:Unavailable
```

- Para configurar o recurso lógico denominadomyApplication offline, digite o seguinte comando:

```
resource.bat -usr john -pwd BXVFDCGS -update -logical myApplication
-setOffline
```

A seguinte saída é exibida:

```
AWKCLI165I Logical resource "myApplication" updated.
```

- Para configurar o computador denominado`myComputer` offline, digite o seguinte comando:

```
resource.bat -usr john -pwd
BXVFDCGS -update -computer myComputer
-setOffline
```

A seguinte saída é exibida:

```
AWKCLI165I Computer "myComputer" updated.
```

- Para recuperar as propriedades básicas do computador denominado`myComputer`, digite o seguinte comando:

```
resource.bat -usr john -pwd BXVFDCGS -query -computer myComputer
```

A seguinte saída é exibida:

```
AWKCLI171I Calling the resource repository to perform a query on resources.
AWKCLI174I "1" computers were found for your query.
Details are as follows:


Computer Name: myComputer
Computer ID:D656470E8D76409F9F4FDEB9D764FF59
Computer OS Name: Microsoft Windows XP Professional English (United States) version
Computer OS Type:Windows XP
Computer OS Version:5
Computer Status:Offline
Computer Availability Status:Unavailable
```

- Para recuperar as propriedades detalhadas do computador denominado`myComputer`, digite o seguinte comando:

```
resource.bat -usr john -pwd BXVFDCGS -query -computer myComputer -v
```

A seguinte saída é exibida:

```
AWKCLI171I Calling the resource repository to perform a query on resources.
AWKCLI174I "1" computers were found for your query.
Details are as follows:


Computer Name: myComputer
Computer ID:D656470E8D76409F9F4FDEB9D764FF59
Computer OS Name:Microsoft Windows XP Professional English (United States) version
Computer OS Type:Windows XP
Computer OS Version:5
Computer Status:Offline
Computer Availability Status:Unavailable
Computer details:
         Physic memory = 2095536.0
         Virtual memory = 3513788.0
         Cpu utilization = 16.0
         Free physic memory = 947972.0
         Free virtual memory = 2333484.0
         Free swap space = 52.0
         Allocated physic memory = 0.0
         Allocated virtual memory = 0.0
         Allocated swap space = 0.0
         Processors number = 1.0
         Allocated processors number = 0.0
         Processor type = x86
         Processor speed = 1995.00
         Manufacturer = IBM
         Model = 2668F8G
         Serial number = L3WZYNC
```

- Para recuperar as propriedades detalhadas do recurso lógico denominado`geneva`, incluindo a lista de computadores associados, digite o seguinte comando:

```
resource.bat -usr
john -pwd BXVFDCGS -query -logical geneva -v
```

  A seguinte saída é exibida:

```
Setting CLI environment variables....
AWKCLI171I Calling the resource repository to perform a query on resources.
AWKCLI172I "1" logical resources were found for your query.
Details are as follows:


Resource Name:geneva
Resource Type:prod_wks
Resource Status:Online
Resource Quantity:1
Resource Current Allocation:0
Computers List:
    Computer Name:bd_ff139_1
    Computer ID:666AADE61CBA11E0ACBECD0E6F3527DE
    Computer Status:Online
    Computer Availability Status:Available
    AWKCLI171I Calling the resource repository to perform a query on resources.
```

- Para criar um grupos de recursos denominado`myGroup`, digite o seguinte comando:

```
resource.bat -usr john -pwd BXVFDCGS -create -group myGroup
```

  A seguinte saída é exibida:

```
AWKCLI153I Resource group "myGroup" created.
```

- Para recuperar propriedades básicas de um grupo de recursos denominado`myGroup`, digite o seguinte comando:

```
resource.bat -query -group myGroup
```

  A seguinte saída é exibida:

```
Setting CLI environment variables....
  AWKCLI171I Calling the resource repository to perform a query on resources.
  AWKCLI173I "1" groups were found for your query.
  Details are as follows:


  Group Name:myGroup
  Group Status:Online
```

- Para incluir o computador denominado`myComputer` em um grupo de recursos denominado`myGroup`, digite o seguinte comando:

```
resource.bat -update -group myGroup -addComputer myComputer
```

  A seguinte saída é exibida:

```
Setting CLI environment variables....
  AWKCLI165I Resource Group "myGroup" updated.
```

- Para recuperar os detalhes de um grupo de recursos denominado`myGroup`, digite o seguinte comando:

```
resource.bat -query -group myGroup -v
```

  A seguinte saída é exibida:

```
Setting CLI environment variables....
  AWKCLI171I Calling the resource repository to perform a query on resources.
  AWKCLI173I "1" groups were found for your query.
  Details are as follows:
```

```
Group Name:myGroup
Group Status:Online
Computers List:
 Computer Name:myComputer
         Computer ID:D656470E8D76409F9F4FDEB9D764FF59
         Computer Status:Online
 Computer Availability Status:Unavailable

Lista de Recursos:
```

Utilizando o comando resource a partir de um agente

É possível criar e gerenciar recursos e grupos de recursos, e computadores de agentes do Tivoli Workload Scheduler que não estejam no gerenciador de domínio principal.

Ativando o Recurso de Auditoria

Para definir o , é necessário:

1. Incluir o tempo de execução para tarefas Java Consulte as informações sobre como instalar o agente no manual *Planejamento e Instalação*.
2. Configure o arquivo **CLIConfig.properties**. Consulte a seção sobre **CLIConfig.properties**.
3. Execute o comando **resource**. Consulte o "Executando o Comando resource" na página 555.

Para esse propósito, uma instância adicional do arquivo **CLIConfig.properties** é instalada em cada agente. Se você pretende executar o comando **resource** a partir de um agente, deverá configurar o**CLIConfig.properties** localmente.

Configurando o Arquivo local.properties

Ao instalar o agente, uma cópia local de **CLIConfig.properties** é automaticamente instalada e parcialmente configurada em seu agente no seguinte caminho:

TWA_home/TWS/TDWB_CLI/config

Para executar o comando **resource.bat** ou **resource.sh** a partir do agente, customize as seguintes palavras-chave do arquivo **CLIConfig.properties**:

ITDWBServerHost
Especifique o endereço IP ou o nome do host do gerenciador de domínio principal.

ITDWBServerPort
Especifica o número da porta WebSphere Application Server.

ITDWBServerSecurePort
Especifica o número da porta WebSphere Application Server.

tdwb_user
Especifica o nome de usuário para um usuário autorizado a executar operações no IBM Tivoli Workload Scheduler quando a segurança estiver ativada. Essa senha deve ser definida previamente no IBM WebSphere. Para obter informações adicionais sobre as considerações de segurança, consulte o *Tivoli Workload Scheduler: Administration Guide*, *SC23-9113*.

tdwb_pwd
Especifica a senha para um usuário autorizado a executar operações no

IBM Tivoli Workload Scheduler quando a segurança estiver ativada. Essa senha deve ser definida previamente no IBM WebSphere. Para obter informações adicionais sobre segurança, consulte o *Tivoli Workload Scheduler: Administration Guide*.

Executando o Comando resource

Dependendo de seu sistema operacional, para executar o comando, insira:

No Windows

```
resource.bat
```

No UNIX

```
resource.sh
```

Alternando Gerenciadores

É possível definir no arquivo `CLIConfig.properties` os servidores broker de backup a serem contatados pela CLI de recurso, se o servidor broker atual não responder. Para configurar a CLI de recurso para entrar em contato com os servidores de backup em caso de falha, você deve especificar no arquivo `CLIConfig.properties` as propriedades de conexão para cada servidor broker de backup. Liste as mesmas propriedades especificadas para o servidor broker em execução no gerenciador de domínio principal primário.

Especifique as seguintes propriedades de conexão:

```
ITDWBServerHost
ITDWBServerPort
ITDWBServerSecurePort
use_secure_connection
tdwb_user
tdwb_pwd
```

Para servidores de backup, o mesmo número ordinal deve ser anexado a cada nome da propriedade associado ao mesmo servidor de backup.

No exemplo a seguir, no arquivo `CLIConfig.properties`, é especificado o servidor broker em execução no gerenciador de domínio principal primário e dois servidores broker de backup:

```
# Propriedades do Servidor Broker em execução no gerenciador de domínio principal primário
ITDWBServerHost = BrokerServer.mycompany.com
ITDWBServerPort = 51117
ITDWBServerSecurePort = 51118
use_secure_connection = true
tdwb_user = tdwbUser
tdwb_pwd = xxxx

# Propriedades do Primeiro (_1) Servidor Broker de Backup
ITDWBServerHost_1 = FirstBackupBrokerServer.mycompany.com
ITDWBServerPort_1 = 41117
ITDWBServerSecurePort_1 = 41118
use_secure_connection_1 = false
tdwb_user_1 = backup1TdwbUser
tdwb_pwd_1 = yyyy

# Propriedades do Segundo (_2) Servidor Broker de Backup
ITDWBServerHost_2 = SecondBackupBrokerServer.mycompany.com
ITDWBServerPort_2 = 61117
ITDWBServerSecurePort_2 = 61118
use_secure_connection_2 = false
tdwb_user_2 = backup2TdwbUser
tdwb_pwd_2 = zzzz
```

É possível definir um máximo de 10 servidores broker.

Para evitar que a CLI de recurso entre em contato com servidores indisponíveis, o nome do último servidor broker contatado com êxito é salvo na propriedade `ITDWBLastGoodServerHost` do arquivo `CLIConfig.properties`.

sendevent

O comando envia a partir de um agente dinâmico ou gerenciador de domínio os eventos customizados definidos com o comando evtdef para o servidor de processador de evento atualmente ativo no plano de produção. À medida que os eventos são recebidos pelo processador de eventos, acionam as regras de eventos nas quais foram especificados.

Nota: No Windows 2012, o comando não é suportado no Windows PowerShell.

Os usuários podem substituir o servidor de destino padrão (definido pelas opções globais) especificando o nome do host e a porta de um novo servidor.

Sintaxe

sendevent -V | ? | -help | -u | -usage

sendevent [**-hostname** *hostname*]
[**-port** *port*]
eventType
source
[[*attribute*=*value*]...]

Argumentos

-V Exibe a versão do comando e sai.

? | -help | -u | -usage
Exibe informações de utilização do comando e sai.

-hostname *hostname*
Especifica o nome do host de um servidor do processador de evento alternativo diferente do atualmente ativo.

-port *port*
Especifica o número da porta de um servidor do processador de eventos alternativo diferente do atualmente ativo.

eventType
Um dos tipos de evento customizado, definido com o comando evtdef no provedor de eventos genéricos e especificado como o evento acionador em uma definição de regra de evento.

origem O nome do provedor de eventos que você customizou com evtdef. Esse também é o nome que você deve especificar como o argumento para a palavra-chave `eventProvider` na definição das regras de evento acionadas por esses eventos customizados.

O nome padrão é `GenericEventPlugIn`.

attribute=*value*
Um ou mais dos atributos qualificadores do tipo de evento customizado que estão especificados como atributos do evento acionador para a regra de evento.

Comentários

O comando neste formulário se aplica ao ambiente dinâmico apenas. Para enviar eventos de agentes não dinâmicos, consulte “sendevent” na página 524.

Exemplos

Neste exemplo, um aplicativo em execução em um agente dinâmico envia o tipo de evento customizado `BusProcCompleted` ao processador de eventos padrão. O evento é que o processamento do arquivo `calcweek` foi concluído.

```
sendevent BusProcCompleted GenericEventPlugIn TransacName=calcweek Workstation=acagn002
```

O nome do arquivo e a estação de trabalho associada são os dois atributos do evento `BusProcCompleted` que foram especificados como atributos de evento acionadores em uma regra de evento associada.

twstrace

Altera o nível de rastreio no agente dinâmico sem ter de parar e reiniciar o agente.

Nota: No Windows 2012, o comando não é suportado no Windows PowerShell.

Autorização

Você deve efetuar login com as credenciais do usuário que instalou oagente dinâmico. Também é possível usar qualquer autorização mais alta que o usuário que instalou oagente dinâmico.

Sintaxe

twstrace -u | -V -enable | -disable [**-level** *value*] [**-maxFiles***files_number*] [**-maxFileBytes** *bytes_number*] [**-getLogs** [**-zipFile** *zip_file_name*] [**-host***hostname*] [**-protocol** *{http|https}*] [**-port** *port number*] [**-inifile** *ini_filename*]]

Argumentos

enable
Ativa o rastreio para o nível máximo. O nível máximo é **3000**. Por padrão, os rastreios são desativados.

disable
Desativa o rastreio.

level *value*
O nível de detalhes para os rastreios:

1000 Mensagens de erro, aviso e informativas são rastreadas.

2000 Mensagens de erro e aviso são rastreadas.

3000 Mensagens de erro são rastreadas.

maxFiles *files_number*
O número máximo dos arquivos de rastreio que deseja criar.

maxFileBytes *bytes_number*
O tamanho máximo em bytes que o arquivo de rastreio pode atingir. O padrão é **1024000** bytes.

getLogs
Para coletar os arquivos de rastreio, os arquivos de mensagens e os arquivos de configuração em um arquivo compactado.

zipfile *zip_file_name*
O nome do arquivo compactado que contém todas as informações,

que são arquivos de logs, de rastreios e de configuração (`ita.ini` e `jobManager.ini`) para o agente. O padrão é `logs.zip`.

host *hostname*
O nome do host ou o endereço IP do agente para o qual deseja coletar os rastreios. O padrão é **localhost**.

protocol *http|https*
O protocolo do agente para o qual você está coletando os rastreios. O padrão é o protocolo especificado no arquivo **.ini** do agente.

port *port number*
A porta do agente. O padrão é o número da porta do agente no qual você está executando a linha de comandos.

inifile *ini_filename*
O nome do arquivo **.ini** que contém a configuração SSL do agente para o qual deseja coletar os rastreios. O padrão é o arquivo **.ini** do agente local. Se estiver coletando o rastreio para um agente remoto para o qual os certificados de segurança foram customizados, você deve importar o certificado no agente local e especificar o nome do arquivo **.ini** que contém a configuração. Para isso, execute as seguintes ações:

1. Extraia o certificado do keystore do agente remoto.
2. Importe o certificado em um keystore do agente local. É possível criar um keystore ad hoc cujo nome deve ser **TWSClientKeyStore.kdb**.
3. Crie um arquivo **.ini** no qual você especifica:
   - **0** na propriedade **tcp_port** como a seguir:
     ```
     tcp_port=0
     ```
   - A porta do agente remoto na propriedade **ssl_port** como a seguir:
     ```
     ssl_port=<ssl_port>
     ```
   - O caminho para o keystore criado na etapa 2. Na propriedade **key_repository_path** a seguir:
     ```
     key_repository_path=<local_agent_keystore_path>
     ```

u Exibe o uso do comando.

V Exibe a versão do produto.

Exemplos

Para configurar o nível de rastreio para registrar mensagens de erro e aviso, execute o seguinte comando:

```
twstrace -enable -level 2000
```

Para recuperar as informações sobre o nível de rastreio, execute o comando a seguir:

```
twstrace -level -maxFiles
-maxFileBytes

AWSITA1761 As propriedades de rastreio são: level="1000",
maxFiles="3", file size="1024000"
```

Capítulo 15. Obtendo Relatórios e Estatísticas

Este capítulo descreve os comandos de relatório que você utiliza para obter um resumo ou informações detalhadas sobre o plano de produção anterior ou seguinte. Esses comandos são executados a partir do prompt de comandos do sistema operacional no gerenciador de domínio principal. Este capítulo está dividido nas seções a seguir:

- "Configuração para Utilizar Comandos de Relatório"
- "Descrições dos comandos" na página 560
- "Amostra de saídas do relatório" na página 568
- "Programas de Extração de Relatório" na página 578
- "Executando Relatórios do Dynamic Workload Console e relatórios em lote" na página 587
- "Executando o relatórios em lote a partir da Interface da Linha de Comandos" na página 591

Configuração para Utilizar Comandos de Relatório

Para configurar o ambiente para utilizar os comandos de relatório, configure as variáveis *PATH* e *TWS_TISDIR* executando um dos seguintes scripts:

- **. ./***TWS_home***/tws_env.sh** para shells de Bourne e Korn no UNIX
- **. ./***TWS_home***/tws_env.csh** para shells C no UNIX
- *TWS_home***tws_env.cmd** no Windows

Os comandos de relatório devem ser executados a partir do diretório *TWS_home*.

A saída dos comandos de relatório é controlada pelas seguintes variáveis de ambiente:

MAESTROLP
: Especifica o destino da saída de um comando. O padrão é **stdout**. É possível defini-lo como qualquer um dos seguintes itens:

  nome do arquivo
  : Grava a saída em um arquivo.

  > *nome do arquivo*
  : Somente para o UNIX. Redireciona a saída para um arquivo, sobrescrevendo o conteúdo do arquivo. Se o arquivo não existir, ele será criado.

  >> *nome do arquivo*
  : Somente para o UNIX. Redireciona a saída para um arquivo, anexando-a ao final do arquivo. Se o arquivo não existir, ele será criado.

  | *command*
  : Somente para o UNIX. Canaliza a saída para um comando do sistema ou processo. O comando do sistema é sempre executado.

© Copyright IBM Corp. 1999, 2014

| | *command*

Somente para o UNIX. Canaliza a saída para um comando do sistema ou processo. O comando do sistema não será executado, se não houver saída.

MAESTRO_OUTPUT_STYLE

Especifica o estilo de saída para nomes de objetos longos. Com valor **LONG**, campos de comprimento integral (longo) são usados para nomes de objeto.

Se a variável estiver configurada para algo que não seja **LONG**, nomes longos são truncados para oito caracteres e um sinal de mais. Por exemplo: **A1234567+**.

Você deve usar uma fonte de tamanho fixo para obter o formato correto das saídas de relatório.

Alterando o formato de data

No Tivoli Workload Scheduler, o formato de data afeta todos os comandos que aceitam uma data como opção de entrada (exceto o comando **datecalc**) e os cabeçalhos em todos os relatórios. O formato de data padrão é*mm/dd/aa*. Para selecionar um formato diferente, edite a opção local ***date format*** armazenada no arquivo `localopts`. Os valores são:

Tabela 81. Formatos de Data

Valor do *formato de data*	**Saída de formato de data correspondente**
0	aa/mm/dd
1	mm/dd/aa
2	dd/mm/aa
3	Variáveis de suporte do idioma nativo.

Consulte o *Guia de Administração do IBM Tivoli Workload Scheduler* para obter detalhes sobre como modificar variáveis locais no arquivo `localopts`.

Descrições dos comandos

Os comandos de relatório do Tivoli Workload Scheduler estão listados na Tabela 82:

Tabela 82. Lista de Comandos de Relatório

Comando	**Descrição**
rep1	Relatório 01 - Listagem de Detalhes do Job
rep2	Relatório 02 - Listagem do Prompt
rep3	Relatório 03 - Listagem de Calendário
rep4a	Relatório 04A - Listagem de Parâmetro
rep4b	Relatório 04B - Listagem de Recurso
rep7	Relatório 07 - Listagem do Histórico do Job
rep8	Relatório 08 - Histograma do Job
rep11	Relatório 11 - Agendamento de Produção Planejada
reptr	Relatório 09A - Resumo de Produção Planejada Relatório 09B - Detalhe de Produção Planejada Relatório 09D - Detalhe de Produção Planejada (Nomes Longos) Relatório 10A - Resumo de Produção Real Relatório 10B - Detalhe de Produção Real
xref	Relatório 12 - Relatório de Referência Cruzada

rep1 - rep4b

Estes comandos imprimem os seguintes relatórios:

Relatório 01
Listagem de Detalhes do Job

Relatório 02
Listagem de Prompts

Relatório 03
Listagem de Calendários

Relatório 04A
Listagem de Parâmetros

Relatório 04B
Listagem de Recursos

Sintaxe

rep[*x*] [**-V**|**-U**]

Execute o comando no diretório *TWS_home*.

Para `rep3`, execute o comando a partir de um diretório ao qual você tem acesso `write`.

Ao imprimir relatórios para tipos de tarefas com opções avançadas, o campo de arquivo JCL retorna o nome do aplicativo.

Argumentos

x Um número correspondente ao relatório. Os números são: **1**, **2**, **3**, **4a** ou **4b**.

-U Exibe as informações de uso do comando e sai.

-V Exibe a versão do comando e sai.

Comentários

A Listagem de Detalhes da Tarefa (relatório 01) não pode incluir tarefas que tenham sido enviadas com a utilização de um nome de alias.

O tempo decorrido exibido para uma tarefa sombra é o tempo decorrido da tarefa remota ao qual ele é limitado.

Exemplos

Relatório de Impressão 03, Listagem do Calendário do Usuário:

```
rep3
```

Exibe informações sobre utilização do comando **rep2**:

```
rep2 -U
```

No UNIX, imprima duas cópias do relatório 04A, Listagem de Parâmetros do Usuário, na impressora `lp2`:

```
MAESTROLP="| lp -dlp2 -n2"
export MAESTROLP
rep4a
```

Este é um relatório de amostra para a tarefa WAGES2_1:

```
Tarefa: WAGES2_1  #FTP
Descrição:
JCL File   : filetransfer
Logon   :          Creator:
tws86
Job de Recuperação:
Tipo de Recuperação : STOP
Aviso de Recuperação :
Composer Autodoc  : Sim
Total de Execuções  :  0 -  0  Bem-sucedidas  0 Interrompidas

                 Decorrido(min)     CPU(seg)
Total    0      0
Normal   0
Última Execução 0      0 (On  0 em  0)
Máximo   0      0 (On  0)
Mínimo   0      0 (On  0)>
```

rep7

Este comando imprime o Relatório 07-Listagem de Históricos do Job.

Sintaxe

rep7 -V | -U

rep7
[**-c** *wkstat*]
[**-s** *jstream_name*]
[**-j** *job*]
[**-f** *date*]
[**-t** *date*]
[**-l**]

Execute o comando no diretório *TWS_home*.

Argumentos

-U Exibe as informações de uso do comando e sai.

-V Exibe a versão do comando e sai.

-c *wkstat*
Especifica o nome da estação de trabalho na qual os jobs são executados. O padrão é todas as estações de trabalho.

-s *jstream_name*
Especifica o nome do fluxo de tarefas no qual as tarefas são executadas. O padrão são todos os fluxos de tarefas.

-j *job* Especifica o nome do job. O padrão são todos os jobs.

-f *date* Especifica que o histórico do job deve ser impresso desta data em diante. Digite a data como *aaaammdd*. O padrão é a data disponível mais antiga.

-t *date* Especifica que o histórico do job deve ser impresso até esta data. Digite a data como *aaaammdd*. O padrão é a data mais recente.

-l Limita as informações da linha de resumo aos jobs que caem no intervalo de data especificado pelas opções -f ou -t. A utilização dessa opção faz com que a ordem de saída seja invertida: a linha de resumo do job será impressa depois das linhas de execução de job individuais. Essa opção será válida somente se você também especificar pelo menos uma das opções -f ou -t.

Comentários

O tempo decorrido exibido para uma tarefa sombra é o tempo decorrido da tarefa remota ao qual ele é limitado.

A qualquer momento que você executar **rep7**, a saída do comando contém as informações armazenadas até a última execução de **JnextPlan**, as informações relacionadas à execução do plano de produção atual estarão contidas na saída **rep7** da próxima vez que você executar **JnextPlan**. Por essa razão, se você executar **rep7** após ter gerado o plano de produção pela primeira vez ou após um comando **ResetPlan**, a saída do comando não contém nenhuma informação de estatística.

Exemplos

Imprima todo o histórico do job para a estação de trabalhoux3:

```
rep7 -c ux3
```

Imprima todo o histórico do job para todos os jobs no fluxo de job `sked25`:

```
rep7 -s sked25
```

Imprima o histórico de job de todos os jobs no fluxo de job `mysked` na estação de trabalhox15 entre 21/1/2005 e 25/1/2005:

```
rep7 -c x15 -s mysked -f 20050121 -t 20050125
```

rep8

Este comando imprime o Relatório 08-Histograma do Job.

Sintaxe

rep8 -V | -U

rep8
[**-f** *date* **-b** *time* **-t** *date* **-e** *time*]
[**-i** *file*]
[**-p**]

rep8
[**-b** *time* **-e** *time*]
[**-i** *file*]
[**-p**]

Execute o comando no diretório *TWS_home*.

Argumentos

-U Exibe as informações de uso do comando e sai.

-V Exibe a versão do comando e sai.

-f *date* Especifica que o histórico do job deve ser impresso desta data em diante. Digite a data como *aaaammdd*. O padrão é a data de hoje.

-b *time*
Especifica que o histórico do job deve ser impresso desta data em diante. Digite a hora como *hhmm*. O padrão é o*startOfDay* do Tivoli Workload Scheduler.

-t *date* Especifica que o histórico do job deve ser impresso até esta data. Digite a data como *aaaammdd*. O padrão é a data mais recente.

-e *time* Especifica que o histórico do job deve ser impresso até esta hora. Digite a hora como *hhmm*. O padrão é o início da hora do dia do Tivoli Workload Scheduler.

-i *file* Especifica o nome do arquivo de log a partir do qual o histórico é extraído. Observe que os arquivos são armazenados no diretório `schedlog`. O padrão é o arquivo `Symphony` atual.

Nota: Assegure que o intervalo de tempo especificado pelos argumentos [**-f** *date* **-b** *time* **-t** *date* **-e** *time*] esteja dentro do intervalo de data e hora definido no nome do arquivo de registro**-i** *file*.

-p Especifica para inserir uma quebra de página após cada data de execução.

Comentários

A qualquer momento que você executar **rep8**, a saída do comando contém as informações armazenadas até a última execução de **JnextPlan**, as informações relacionadas à execução do plano de produção atual estarão contidas na saída **rep8** da próxima vez que você executar **JnextPlan**. Por essa razão, se você executar **rep8** após ter gerado o plano de produção pela primeira vez ou após um comando **ResetPlan**, a saída do comando não contém nenhuma informação de estatística.

Exemplos

Imprima um histograma do job que inclua todas as informações no plano atual (arquivo `Symphony`):

```
rep8
```

Imprima um histograma do job que comece às 6h em 25/1/2005 e termine às 5h59 em 26/1/2005:

```
rep8 -f 20050125 -b 0600 -t 20050126 -e 0559 -i schedlog/M199801260601
```

Imprima um histograma do job a partir do plano atual (arquivo `Symphony`), que comece às 6h e termine às 22h:

```
rep8 -b 0600 -e 2200
```

rep11

Este comando imprime o Relatório 11-Agendamento de Produção Planejada.

Sintaxe

rep11 -V|-U

rep11
[**-m** *mm*[*yy*] [...]]
[**-c** *wkstat* [...]]
[**-s** *jstream_name*]
[**-o** *output*]

Execute o comando no diretório *TWS_home*.

Argumentos

-U Exibe as informações de uso do comando e sai.

-V Exibe a versão do comando e sai.

-m *mm*[*yy*]
Especifica os meses a serem relatados. Digite o número do mês como *mm*. O padrão é o mês atual.

Você também pode digitar um ano como *yy*. O padrão será o ano atual ou o próximo ano, se você especificar um mês anterior ao mês atual.

-c *wkstat*
Especifica as estações de trabalho a serem relatadas. O padrão é todas as estações de trabalho.

-s *jstream_name*
Especifica o nome do fluxo de tarefas no qual as tarefas são executadas. O padrão são todos os fluxos de tarefas.

-o *output*
Especifica o arquivo de saída. O padrão é o arquivo definido pela variável ***MAESTROLP***. Se ***MAESTROLP*** não for definido, o padrão será **stdout**.

Exemplos

Relate em junho, julho e agosto de 2004 para as estações de trabalho`main`, `site1` e `sagent1`:

```
repll -m 0604 0704 0804 -c main site1 sagent1
```

Relate em Junho, Julho e Agosto deste ano para todas as estações de trabalho e direcione a saída para o arquivo `rllout`:

```
repll -m 06 07 08 -o rllout
```

Relate neste mês e ano para a estação de trabalho`site2`:

```
repll -c site2
```

reptr

Este comando imprime os seguintes relatórios:

Relatório 09A
Resumo da Produção Planejada

Relatório 09B
Detalhe da Produção Planejada

Relatório 10A
Resumo da Produção Real

Relatório 10B
Detalhe da Produção Real

Relatório 09A e Relatório 09B referem-se a processamento de produção futuro, enquanto que Relatório 10A e relatório 10B mostram os resultados e o status de cada tarefa única de produção já processada.

Sintaxe

reptr [**-V**|**-U**]

reptr -pre
[**-**{**summary** | **detail**}]
[*symfile*]

reptr -post
[-{**summary** | **detail**}]
[*logfile*]

Execute o comando a partir de um diretório ao qual você tem acesso `write`.

Argumentos

-U Exibe as informações de uso do comando e sai.

-V Exibe a versão do comando e sai.

-pre Especifica que os relatórios de pré-produção devem ser impressos (09A e 09B).

-post Especifica que os relatórios de pós-produção devem ser impressos (10A e 10B).

-summary
Especifica que os relatórios de resumo devem ser impressos (09A e 10A). Se **-summary** e **-detail** forem omitidos, ambas as definições de relatórios serão impressas.

-detail Especifica que os relatórios de detalhes devem ser impressos (09B e 10B). Se **-summary** e **-detail** forem omitidos, ambas as definições de relatórios serão impressas.

symfile Especifica o nome do arquivo de plano a partir do qual os relatórios serão impressos. O padrão é `Symnew` no diretório atual. Se o arquivo não estiver no diretório atualmente em funcionamento, você deve incluir o caminho absoluto para o nome do arquivo.

logfile Especifica o nome completo do arquivo de log a partir do qual os relatórios serão impressos. Observe que os arquivos de log de plano são armazenados no diretório `schedlog`. O padrão é plano atual (arquivo `Symphony`).

Se o comando for executado sem opções, os dois relatórios **pré** (09A e 09B) serão impressos e as informações serão extraídas do arquivo `Symphony`.

Exemplos

Imprima o relatório de detalhe de produção anterior a partir do arquivo `Symnew`:

```
reptr -pre -detail
```

Imprima o relatório de resumo de produção anterior a partir do arquivo `mysym`:

```
reptr -pre -summary mysym
```

Imprima o relatório de resumo de produção posterior a partir do arquivo de log `M199903170935`:

```
reptr -post -summary schedlog/M199903170935
```

Imprima os relatórios pré-produção lendo a partir do arquivo Symphony.

```
reptr
```

Quando os argumentos forem especificados, os relatórios pré-produção serão baseados nas informações lidas a partir do arquivo `Symnew` enquanto os relatórios pós-produção serão baseados nas informações lidas a partir do arquivo `Symphony`.

xref

Este comando imprime o Relatório 12-Relatório de Referência Cruzada.

Sintaxe

xref [**-V** | **-U**]

xref
[**-cpu** *wkstat*]
[**-depends** | **-files** | **-jobs** | **-prompts** | **-resource** | **-schedules** | **-when**[...]]

Execute o comando no diretório *TWS_home*.

Argumentos

-U Exibe as informações de uso do comando e sai.

-V Exibe a versão do comando e sai.

-cpu *wkstat*
Especifica que o relatório para a estação de trabalho nomeado deve ser impresso. O caractere curinga @ é permitido, neste caso, as informações de todas as estações de trabalho qualificadas são incluídas. O padrão é todas as estações de trabalho.

-depends
Especifica que um relatório mostrando os fluxos de job e jobs que são sucessores de cada job deve ser impresso.

-files Especifica que um relatório mostrando os fluxos de job e jobs que são dependentes de cada arquivo deve ser impresso.

-jobs Especifica que um relatório mostrando os fluxos de job nos quais cada job é executado deve ser impresso.

-prompts
Especifica que um relatório mostrando os fluxos de job e jobs que são dependentes em cada prompt deve ser impresso.

-resource
Especifica que um relatório mostrando os fluxos de job e jobs que são dependentes em cada recurso deve ser impresso.

-schedules
Especifica que um relatório mostrando os fluxos de job e jobs que são sucessores de cada fluxo de job deve ser impresso.

-when Especifica que um relatório mostrando as datas de `Inclusão` e `Exclusão` do fluxo de job deve ser impresso.

Se o comando for executado sem nenhuma opção, todas as estações de trabalho e todas as opções serão selecionadas.

Exemplos

Imprima um relatório para todas as estações de trabalho, mostrando todas as informações sobre referência cruzada:

```
xref
```

Imprima um relatório para todas as estações de trabalho. Inclua informações de referência cruzada sobre todas as dependências do sucessor:

```
xref -cpu @ -depends -schedules
```

Amostra de saídas do relatório

Relatório 01 - Listagem de Detalhes da Tarefa:

```
TWS para UNIX (AIX)/REPORT1 8.3 (1.7)   ibm       Página 1
Relatório 01       Listagem de Detalhes de Tarefas  06/03/06

Job    : FTAWIN8+  #SCHEDDDD
Descrição    :
Arquivo JCL   : dir
Logon     : maestro_adm
Criador    : raiz
Job de Recuperação:
Tipo de Recuperação : STOP
Aviso de Recuperação :
composer Autodoc  : Sim
Total de Execuções  :  0 -  0  Bem-sucedidas  0 Interrompidas

                 Decorrido(min)  CPU(seg)
Total    00:00:00    0
Normal   00:00:00
Última Execução 00:00:00    0 (Em   às 00:00)
Máximo   00:00:00    0 (Em   )
Mínimo   00:00:00    0 (Em   )


Job    : MASTER8+  #JnextPlan
Descrição    : INCLUÍDO pelo composer PARA PLANEJAMENTO MASTER821#FINAL.
Arquivo JCL  : /test/maestro_adm/tws/JnextPlan
Logon     : maestro_adm
Criador    : maestro_adm
Job de Recuperação:
Tipo de Recuperação : STOP
Aviso de Recuperação :
composer Autodoc  : Sim
Total de Execuções  :  11 -  11  Bem-sucedidas  0 Interrompidas

                 Decorrido(min)  CPU(seg)
Total    00:00:14    44
Normal   00:00:01
Última Execução 00:00:01    4 (Em 05/03/06 às 23:16)
Máximo   00:00:02    4 (Em 04/03/06)
Mínimo   00:00:01    4 (Em 04/03/06)


Job    : MASTER8+  #JOB1
Descrição    : INCLUÍDO pelo composer.
Arquivo JCL   : pwd
Logon     : ^ACCLOGIN^
Criador    : raiz
Job de Recuperação:
Tipo de Recuperação : STOP
Aviso de Recuperação :
composer Autodoc  : Sim
Total de Execuções  :  1 -  1  Bem-sucedidas  0 Interrompidas

                 Decorrido(min)  CPU(seg)
Total    00:00:01    0
Normal   00:00:01
Última Execução 00:00:01    0 (Em 05/03/06 às 22:22)
Máximo   00:00:01    0 (Em 05/03/06)
Mínimo   00:00:01    0 (Em 05/03/06)

                    * * * * F i m  d o  R e l a t ó r i o * * * *
```

Na saída, você vê os valores definidos em “Tarefa” na página 711 da seguinte forma:

composer Autodoc
Informa se a instrução do job foi descrita na definição do fluxo de job utilizando a interface de linha de comandos.

CPU (seg)
É o tempo real da CPU, expresso em segundos, utilizado para a execução do job.

Total É a soma do tempo de CPU gravada para o 'Total de Execuções'.

Normal
É o valor médio de tempo da CPU gravado durante o 'Total de Execuções'.

Última Execução
É o tempo de CPU gravado durante a última execução do job.

Máximo
É o máximo entre os valores coletados para o tempo de CPU durante o 'Total de Execuções' (calculado apenas para tarefas concluídas com êxito).

Mínimo
É o mínimo entre os valores coletados para o tempo de CPU durante o 'Total de Execuções' (calculado apenas para tarefas concluídas com êxito).

Criador
É o nome do usuário que criou a definição do job.

Descrição
É a descrição de texto do job definida no campo **Descrição** da instrução de definição do job.

Decorrido
É o período de tempo, expresso em minutos, que inclui o tempo durante o qual o job utilizou a CPU e o tempo que o job teve que aguardar por outros processos para liberar a CPU.

Total É a soma do tempo Decorrido gravada para o 'Total de Execuções'.

Normal
É o valor médio de tempo Decorrido gravado durante o 'Total de Execuções'.

Última Execução
É o tempo Decorrido gravado durante a última execução do job.

Máximo
É o máximo entre os valores coletados para o Tempo Decorrido durante o 'Total de Execuções' (calculado apenas para tarefas concluídas com êxito).

Mínimo
É o mínimo entre os valores coletados para o Tempo Decorrido durante o 'Total de Execuções' (calculado apenas para tarefas concluídas com êxito).

Nota: O tempo decorrido exibido para uma tarefa sombra é o tempo decorrido da tarefa remota ao qual ele é limitado.

Arquivo JCL
É o nome do arquivo definido no campo **scriptname** que contém o script a ser executado ou o comando especificado no campo **docommand** a ser chamado ao executar o job.

Tarefa É o identificador da tarefa, [*workstation***#**]*jobname*.

Logon É o nome do usuário, especificado no campo **streamlogon**, com o qual o job é executado.

Tarefa de Recuperação
É a tarefa, especificada como *after [workstation#]jobname*, que é executada se a tarefa pai for encerrada de forma anormal.

Prompt de Recuperação
É o texto do aviso, especificado no campo **abendprompt**, que será exibido se esse job terminar de forma anormal.

Tipo de Recuperação
É a opção de recuperação definida na definição do job. Pode ser definida como **stop**, **continue** ou **rerun**.

Relatório 02 - Listagem de Avisos:

```
TWS para UNIX (AIX)/REPORT2 8.3 (1.7)   ibm       Página 1
Relatório 02       Listagem de Mensagens de Prompts  06/03/06


Aviso  Mensagem

PROMPT1  Responder SIM quando estiver pronto para executar acc103 e acc104.
PROMPT2  Todos os usuários efetuaram logout?
CALLNO  555-0911
CALLOPER Chamar ^PERSON2CALL^ en ^CALLNO^ para assegurar que todas as placas de tempo tenham sido processadas.
PERSON2CALL Lou Armstrong

Número total de avisos no arquivo:  5

                        * * * * F i m  d o  R e l a t ó r i o * * * *
```

A saída do Relatório 02 lista o nome e o texto dos avisos definidos no ambiente.

Relatório 03 - Listagem de Calendários:

```
TWS para UNIX (AIX)/REPORT3 8.3 (1.7)   ibm       Página 1
Relatório 03       Lista de Calendários do Usuário  06/03/06


Tipo de calendário: MONTHEND
Descrição: Final do mês até o final de 2006.


          Jan 2006      Fev 2006      Mar 2006
Dom Seg Ter Qua Qui Sex Sáb Dom Seg Ter Qua Qui Sex Sáb Dom Seg Ter Qua Qui Sex Sáb
                         .             .   .   .   .   .             .   .   .   .   .
 .   .   .   .   .   .   .     .   .   .   .   .   .   .     .   .   .   .   .   .   .
 .   .   .   .   .   .   .     .   .   .   .   .   .   .     .   .   .   .   .   .   .
 .   .   .   .   .   .   .     .   .   .   .   .   .   .     .   .   .   .   .   .   .
 .   .   .   .   .   .   .     .  28       .   .   .   .  31
 .  31

          Abr 2006      Mai 2006      Jun 2006
Dom Seg Ter Qua Qui Sex Sáb Dom Seg Ter Qua Qui Sex Sáb Dom Seg Ter Qua Qui Sex Sáb
                     .   .     .   .   .   .   .   .   .                 .   .   .   .
 .   .   .   .   .   .   .     .   .   .   .   .   .   .     .   .   .   .   .   .   .
```

```
 .   .   .   .   .   .   .    .   .   .   .   .   .   .    .   .   .   .   .   .   .
 .   .   .   .   .   .   .    .   .   .   .   .   .   .    .   .   .   .   .   .   .
 .   .   .   .   .   .  30  .   .  31     .   .   .   .  30

          Jul 2006      Ago 2006      Set 2006
Dom Seg Ter Qua Qui Sex Sáb Dom Seg Ter Qua Qui Sex Sáb Dom Seg Ter Qua Qui Sex Sáb
                      .   .       .   .   .   .   .   .                   .   .   .
 .   .   .   .   .   .   .    .   .   .   .   .   .   .    .   .   .   .   .   .   .
 .   .   .   .   .   .   .    .   .   .   .   .   .   .    .   .   .   .   .   .   .
 .   .   .   .   .   .   .    .   .   .   .   .   .   .    .   .   .   .   .   .   .
 .   .   .   .   .   .   .    .   .   .  31     .   .   .   .   .  30
31

          Out 2006      Nov 2006      Dez 2006
Dom Seg Ter Qua Qui Sex Sáb Dom Seg Ter Qua Qui Sex Sáb Dom Seg Ter Qua Qui Sex Sáb
                      .           .   .   .   .   .                   .   .   .
 .   .   .   .   .   .   .    .   .   .   .   .   .   .    .   .   .   .   .   .   .
 .   .   .   .   .   .   .    .   .   .   .   .   .   .    .   .   .   .   .   .   .
 .   .   .   .   .   .   .    .   .   .   .   .   .   .    .   .   .   .   .   .   .
 .   .   .   .   .   .   .    .   .   .  30    .   .   .   .   .   .  31
 .  31

                        * * * * F i m  d o  R e l a t ó r i o * * * *
```

Na saída, você vê destacado o final dos dias do mês selecionados no calendário**MONTHEND**.

Relatório 04A - Listagem de Parâmetros:

```
TWS para UNIX (AIX)/REPORT4A 8.3 (1.7)   ibm       Página 1
Relatório 4A        Listagem de Parâmetros do Usuário    06/03/06


Nome do Parâmetro   Conteúdo

ACCHOME     /usr/local/Tivoli/maestro_adm
ACCLOGIN     maestro_adm
BADEXIT     99
GOODEXIT     0
SCRPATH     /usr/local/Tivoli/maestro_adm/scripts

Número de Parâmetros no arquivo: 5

                        * * * * F i m  d o  R e l a t ó r i o * * * *
```

A saída do Relatório 04A lista o nome e o conteúdo dos parâmetros definidos no ambiente.

Relatório 04B - Listagem de Recursos:

```
TWS para UNIX (AIX)/REPORT4B 8.3 (1.7)   ibm       Página 1
Relatório 4B        Listagem de Recursos do TWS  06/03/06


               Recurso    Número
CPU    Nome     Disponib.  Descrição

FTAHP   #DATTAPES    1   Unidades de fita DAT

FTAWIN8+   #QUKTAPES    2   Unidades de fita Quick

MASTER8+   #TAPES     2   Unidades de fita

MASTER8+   #JOBSLOTS   1024   Slots de tarefas
```

```
Número de Recursos no arquivo:  4

                        * * * * F i m  d o  R e l a t ó r i o * * * *
```

A saída do Relatório 04B lista o nome, o número de recursos disponíveis definidos no ambiente e sua descrição.

Relatório 07 - Listagem de Históricos da Tarefa:

```
TWS para UNIX (AIX)/REPORT7 8.3 (1.13)   ibm      Página 1
Relatório 07       Listagem de Históricos da Tarefa  08/03/06

Data  Hora  Nome do Fluxo de Tarefas Decorrido  CPU  Status

Tarefa:MASTER8+#MyJS Execuções: Interrompidas 0 Bem-sucedidas 11 Tempo Decorrido: Normal 1 Mín. 1 Máx. 2

03/03/06 01:46  MASTER8+#JS1   1   4  SU
03/03/06 19:08  MASTER8+#JS2   1   4  SU
03/03/06 19:33  MASTER8+#JS3   1   4  SU
03/03/06 19:37  MASTER8+#JS4   1   4  SU
03/03/06 23:08  MASTER8+#JS5   2   4  SU
03/03/06 05:59  MASTER8+#JS_A   1   4  SU
03/05/06 05:59  MASTER8+#JS_G   1   4  SU
03/06/06 05:59  MASTER8+#JS_H   1   4  SU
03/06/06 21:57  MASTER8+#TIMEJ   2   4  SU
03/06/06 23:16  MASTER8+#SLEEPJ   1   4  SU

Job:MASTER8+#JOB1  Execuções: Interrompidas 0 Bem-sucedidas 1 Tempo Decorrido: Normal 1 Mín. 1 Máx. 1

03/06/06 22:22  MASTER8+#JOBS   1   0  SU


                        * * * * F i m  d o  R e l a t ó r i o * * * *
```

O Relatório 7 lê as informações sobre a execução da tarefa armazenada no banco de dados e as exibe. Os possíveis estados para uma tarefa são:

AB para tarefas em falha

SU para tarefas concluídas com êxito

DN para tarefas enviadas cujo estado é desconhecido porque nem uma mensagem bem-sucedida nem uma em falha ainda foi recebida.

Relatório 08 - Histograma da Tarefa:

```
TWS para UNIX (AIX)/REPORT8 8.3 (1.7)   ibm       Página 1
Relatório 08     Histograma da Tarefa 05/03/06 14:05 - 06/03/06 14:04  06/03/05

Intervalo Por Coluna: 15 minutos

                        1 1 1 1 2 2 2 0 0 0 0 0 0 0 1 1 1
                        4 5 7 8 0 1 3 0 2 3 5 6 8 9 1 2 4
                        0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0
                        5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 4

Nome do Job  Stat
06/03/06

CF05066+.JnextPlan SU           .*.


                        * * * * F i m  d o  R e l a t ó r i o * * * *
```

A saída do Relatório 8 mostra os períodos de tempo durante os quais os jobs são executados. Os números no início do histograma do job são horários, escritos de cima para baixo, por exemplo, a primeira coluna 1405 significa 14h05. Os períodos de tempo quando o job é executado são marcados por asteriscos quando a posição do marcador é alinhada com um horário escrito de cima para baixo e pontos.

Relatório 9B - Detalhes da Produção Planejada:

```
TWS para UNIX (AIX)/REPORT 8.3 (1.7)   ibm       Página 1
Relatório 09B Symnew  Detalhe da Produção Planejada para 06/03/06     06/03/06

                        Tempo de Exec.
                Nome do Trab.  Estimado  Pri Hora de Início Até Todas as Dep. de Limite

Planejamento NETAG #EXTERNAL   0
                E0000000    0
               Total  00:00
               Total  00:00

Planejamento MYFTA #IWDSKE   10            NETAG#EXTERNAL.E0000000
                JOBIWD   10     23:00(03/06/06) 01:00
               Total  00:00

Planejamento MYMST #TESTSKE 00:29 10
                TESTCRO+ 00:01 10
                NEWTEST 00:29 10 08:30(06/03/06)        TESTCROME
               Total  00:29


Planejamento MYMST #FINAL  00:00 10 05:59(07/03/06)
                JnextPlan 00:01 10
               Total  00:01
               Total  00:34

                          * * * * F i m  d o  R e l a t ó r i o * * * *
```

A saída do Relatório 9B mostra o que está planejado para ser executado na data selecionada no ambiente do Tivoli Workload Scheduler. As informações exibidas são retiradas das definições armazenadas no banco de dados do Tivoli Workload Scheduler. A saída mostra os fluxos de tarefas cuja execução está planejada para o dia 6 de março de 2006 com sua descrição, a lista de tarefas contidas, as dependências de tempo, a taxa de repetição e o limite de tarefa, se configurado, e a dependência de outras tarefas ou fluxos de tarefas. Por exemplo, o fluxo de job chamado`iwdske`, planejado para ser executado em `MYFTA`, tem uma dependência `follows` no job `NETAG#EXTERNAL.E0000000`, planejado para ser executado no agente de rede chamado`NETAG`.

O campo **Hora de Início** na saída dos relatórios gerados pelo comando **reptr** mostra:

Uma restrição de tempo configurada na definição do fluxo de tarefas utilizando a palavra-chave at.
Se a data estiver colocada entre parênteses (), por exemplo:

```
Horário de Início
06:00(20/03/06)
```

A hora em que o fluxo de tarefa está planejado para execução configurada na definição do fluxo de tarefa utilizando a palavra-chave schedtime.
Se a data estiver colocada entre chaves {}, por exemplo:

```
Horário de Início
06:00{03/20/06}
```

A hora em que o fluxo de tarefa realmente iniciou a execução.
Se a data não estiver entre chaves ou entre parênteses, por exemplo:

```
                                Horário de Início
                                06:00 20/03/06
```

Relatório 10B - Detalhes da Produção Real:

```
TWS para UNIX (AIX)/REPORT 8.3 (1.7)   ibm       Página 1
Relatório 10B Symphony   Detalhe da Produção Real para 06/03/06   07/03/06

                         Tempo de Exec.        Tempo de Exec.  CPU -  Status do
                 Nome do Job  Estimado  Prioridade Hora de Início  Real   Núm. de Seg. Job

Planejamento NETAG #EXTERNAL    0            EXTRN
                 E0000000    0            ERROR
                  Total  00:00       00:00 0

Planejamento MYMST #MONTHSKE 00:02 10  06:01(06/03/06) 00:03     SUCC
                 GETLOGS  00:02 10  06:01(06/03/06) 00:03   #J11612 SUCC
                  Total  00:02       00:03 0

Planejamento MYFTA #IWSKE    10            HOLD
                 JOBIWD    10            HOLD
                  Total  00:00       00:00 0

Planejamento MYMST #TESTSKE  00:29 10  06:01(06/03/06) 00:02     STUCK
                 TESTCRO+ 00:01 10  06:01(06/03/06) 00:02   #J11613 ABEND
                 NEWTEST  00:29 10            HOLD
                  Total  00:30       00:02 0

Planejamento MYMST #FINAL  00:01 10  05:59(07/03/06)       HOLD
                 JnextPlan 00:01 10            HOLD
                  Total  00:01       00:00 0


                  Total  01:38       00:09 0

                              * * * * F i m  d o  R e l a t ó r i o * * * *
```

A saída do Relatório 10B mostra os estados das atividades de planejamento atualmente em execução na rede do Tivoli Workload Scheduler. As informações exibidas são retiradas da cópia do arquivo Symphony utilizado atualmente e atualizado no ambiente de planejamento. Isso significa que a qualquer momento que esse comando de relatório for executado durante o processamento, as informações exibidas refletirão o status real da atividade planejada.

Se você comparar essa saída com a saída do Relatório 9B, verá que o fluxo de tarefas MONTHSKE foi executado durante o dia de produção atual, 6 de março, mas sua execução não está planejada para o dia seguinte, 7 de março. O fluxo de job EXTERNAL, por sua vez, falhou no agente de rede NETAG e, assim, o fluxo de job IWSKE, que tem uma dependência follows do fluxo de job EXTERNAL, permanece no estado HOLD.

O fluxo de job TESTSKE, por sua vez, está no estado STUCK, que significa que é necessária a intervenção do operador porque, dentro do tempo de execução do fluxo de job, o job TESTCROME, depois de ter sido iniciado com o ID de job J11613, falhou no estado ABEND, fazendo com que o job dependente NEWTEST mudasse para o estado HOLD.

Relatório 11 Agendamento de Produção Planejada:

```
TWS para UNIX (AIX)/REPORT11 8.3 (1.7)           Página 1
Relatório 11     Agendamento de Produção Planejada para FEV 2006    08/03/06

CPU: FTAWIN8+
```

```
         Núm Tempo 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
Planej. Trab. Est.Cpu Te Qu Qu Se Sá Do Se Te Qu Qu Se Sá Do Se Te Qu Qu Se Sá So Se Te Qu Qu Se Sá Do Se
SCHED1 1  1         *

Um * entre o Nome do Planejamento e o Número de Jobs indica que o planejamento possui jobs em execução em outras cpus.
-------------------------------------------------------------------------------------------------------------
Tempo Estimado de Cpu por Dia em Segundos
     Seg  Ter  Qua  Qui  Sex  Sáb  Dom

                1  2  3  4  5  6
                0  0  0  0  0  0

       7  8  9  10  11  12  13
       0  0  0  0  1  0  0

      14  15  16  17  18  19  20
       0  0  0  0  0  0  0

      21  22  23  24  25  26  27
       0  0  0  0  0  0  0

      28
       0


TWS para UNIX (AIX)/REPORT11 8.3 (1.7)           Página 2
Relatório 11     Agendamento de Produção Planejada para FEV 2006    08/03/06

CPU: MASTER8+

         Núm Tempo 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
Planej. Trab. Est.Cpu Te Qu Qu Se Sá Do Se Te Qu Qu Se Sá Do Se Te Qu Qu Se Sá So Se Te Qu Qu Se Sá Do Se
FINAL  1  4 * * * * * * * * * * * * * * * * * * * * * * * * * * * *

Um * entre o Nome do Planejamento e o Número de Jobs indica que o planejamento possui jobs em execução em outras cpus.
-------------------------------------------------------------------------------------------------------------
Tempo Estimado de Cpu por Dia em Segundos
     Seg  Ter  Qua  Qui  Sex  Sáb  Dom

                1  2  3  4  5  6
                4  4  4  4  4  4

       7  8  9  10  11  12  13
       4  4  4  4  4  4  4

      14  15  16  17  18  19  20
       4  4  4  4  4  4  4

      21  22  23  24  25  26  27
       4  4  4  4  4  4  4

      28
       4


                        * * * * F i m  d o  R e l a t ó r i o * * * *
```

A saída do Relatório 11 mostra quando os fluxos de job são planejados para execução durante o mês selecionado. Na primeira linha é exibido o número de jobs que o fluxo de job contém, o tempo estimado de CPU utilizado pela execução do fluxo de job e para quando a execução do fluxo de job foi planejada. Na matriz, é exibido para cada dia do mês selecionado o tempo estimado de CPU utilizado para a execução desse fluxo de job.

Relatório 12 - Relatório de Referências Cruzadas:

A saída do Relatório 12 mostra informações diferentes de acordo com o sinalizador utilizado ao emitir o comando `xref`. Nesta seção, você localiza algumas amostras de saída. Para cada uma delas, o sinalizador correspondente utilizado com o comando xref é realçado.

xref -when

```
TWS para UNIX (AIX)/CROSSREF 8.3 (1.7)   ibm       Página 1
Relatório 12 Relatório de Referência Cruzada para as opções ON, EXCEPT(*) e FREEDAYS(f). 08/03/06

CPU: FTAHP

WHEN     Utilizado pelos seguintes planejamentos:
REQUEST    TRFINAL



TWS para UNIX (AIX)/CROSSREF 8.3 (1.7)   ibm       Página 2
Relatório 12 Relatório de Referência Cruzada para as opções ON, EXCEPT(*) e FREEDAYS(f). 08/03/06

CPU: FTAWIN8+

WHEN     Utilizado pelos seguintes planejamentos:
MONTHEND    SCHED1
REQUEST    SCHED1 , SCHEDDAA



TWS para UNIX (AIX)/CROSSREF 8.3 (1.7)   ibm       Página 3
Relatório 12 Relatório de Referência Cruzada para as opções ON, EXCEPT(*) e FREEDAYS(f). 08/03/06

CPU: MASTER8+

WHEN     Utilizado pelos seguintes planejamentos:
EVERYDAY    FINAL
REQUEST    TMP


                    * * * * F i m  d o  R e l a t ó r i o * * * *
```

xref -jobs

```
TWS para UNIX (AIX)/CROSSREF 8.3 (1.7)   ibm       Página 4
Relatório 12 Relatório de Referência Cruzada para Nomes de Tarefas.           08/03/06

CPU: FTAWIN8+

Nome da Tarefa  Existe em Planejamentos
SCHEDDDD    SCHED1



TWS para UNIX (AIX)/CROSSREF 8.3 (1.7)   ibm       Página 5
Relatório 12 Relatório de Referência Cruzada para Nomes de Tarefas.           08/03/06

CPU: MASTER8+

Nome do Job  Existe em Planejamentos
JnextPlan    FINAL
JOB1     TMP


                    * * * * F i m  d o  R e l a t ó r i o * * * *
```

xref -resource

```
TWS para UNIX (AIX)/CROSSREF 8.3 (1.7)   ibm       Página 8
Relatório 12                   Relatório de Referência Cruzada para Usuários do Recurso.        08/03/06

CPU: FTAWIN8+

Recurso    Utilizado pelo seguinte:
QUKTAPES(N/F)   SCHED1



TWS para UNIX (AIX)/CROSSREF 8.3 (1.7)   ibm       Página 9
Relatório 12                   Relatório de Referência Cruzada para Usuários do Recurso.        08/03/06

CPU: MASTER8+

Recurso    Utilizado pelo seguinte:
TAPES(N/F)   TMP


                        * * * * F i m  d o  R e l a t ó r i o * * * *
```

xref -prompts

```
TWS para UNIX (AIX)/CROSSREF 8.3 (1.7)   ibm       Página 6
Relatório 12             Relatório de Referência Cruzada para Dependências de Prompt.     08/03/06
CPU: FTAWIN8+

Aviso     Utilizado pelo seguinte:

Texto definido pelo usuário  SCHED1



TWS para UNIX (AIX)/CROSSREF 8.3 (1.7)   ibm       Página 7
Relatório 12             Relatório de Referência Cruzada para Dependências de Prompt.     08/03/06

CPU: MASTER8+

Aviso     Utilizado pelo seguinte:
BADEXIT    FTAWIN8+#SCHED1
GOODEXIT    FTAWIN8+#SCHED1 , TMP

Texto definido pelo usuário  TMP


                        * * * * F i m  d o  R e l a t ó r i o * * * *
```

xref -files

```
TWS para UNIX (AIX)/CROSSREF 8.3 (1.7)   ibm       Página 10
Relatório 12             Relatório de Referência Cruzada para Dependências de Arquivos.      08/03/06

CPU: MASTER8+

Nome do Arquivo  Utilizado pelo seguinte:
/root/MY_FILE.sh  FTAWIN8+#SCHED1 , TMP


                        * * * * F i m  d o  R e l a t ó r i o * * * *
```

Programas de Extração de Relatório

Os programas de extração de relatório são utilizados para gerar vários relatórios do Tivoli Workload Scheduler. Os programas são listados na Tabela 83:

Tabela 83. Programas de Extração de Relatório.

Programa de extração de relatório	Descrição
jbxtract	Utilizado para gerar o Relatório 01 - Listagem de Detalhes da Tarefa e para o Relatório 07 - Listagem de Históricos da Tarefa
prxtract	Utilizado para gerar o Relatório 02 - Listagem de Prompts
caxtract	Utilizado para gerar o Relatório 03 - Listagem de Calendários
paxtract	Utilizado para gerar o Relatório 04A - Listagem de Parâmetros
rextract	Utilizado para gerar o Relatório 04B - Listagem de Recursos
r11xtr	Utilizado para gerar o Relatório 11 - Agendamento de Produção Planejada
xrxtrct	Utilizado para gerar o Relatório 12 - Relatório de Referência Cruzada

A saída dos programas de extração é controlada pela variável ***MAESTRO_OUTPUT_STYLE***, que define por quanto tempo os nomes de objetos serão manipulados. Para obter informações adicionais sobre a variável ***MAESTRO_OUTPUT_STYLE***, consulte "Descrições dos comandos" na página 560.

jbxtract

Extrai informações sobre tarefas do banco de dados.

Sintaxe

jbxtract [**-V** | **-U**]
[**-j** *job*]
[**-c** *wkstat*]
[**-o** *output*]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

-j *job* Especifica o job para o qual a extração é executada. O padrão são todos os jobs.

-c *wkstat*
Especifica a estação de trabalho de jobs para a qual a extração é executada. O padrão é todas as estações de trabalho.

-o *output*
Especifica o arquivo de saída. O padrão é **stdout**.

Resultados

A variável ***MAESTRO_OUTPUT_STYLE*** especifica o estilo de saída para nomes longos de objetos. Com valor **LONG**, campos de comprimento integral (longo) são usados para nomes de objeto. Se a variável estiver configurada para algo que não seja **LONG**, nomes longos são truncados para oito caracteres e um sinal de mais. Por exemplo: `A1234567+`.

Cada registro de job contém campos de comprimento variável, delimitados por tabulação. Os campos são descritos na Tabela 84 na página 579.

Tabela 84. Campos de Saída de Jbxtract

Campo	Descrição	Comprimento Máximo (bytes)
1	nome da estação de trabalho	16
2	nome da tarefa	16
3	nome do arquivo de script do job	4096
4	descrição da tarefa	65
5	nome do job de recuperação	16
6	opção de recuperação (0=stop, 1=rerun, 2=continue)	5
7	texto do prompt de recuperação	64
8	sinalizador de documentação automática (0=desativado, 1=ativado)	5
9	nome do usuário de logon do job	36
10	nome de usuário do criador do job	36
11	número de execuções bem sucedidas	5
12	número de execuções terminadas de forma anormal	5
13	tempo total decorrido de todas as execuções do job	8
14	tempo total de cpu de todas as execuções do job	8
15	tempo decorrido médio	8
16	última data de execução (ddmmyy)	8
17	última hora de execução (hhmm)	8
18	últimos segundos da cpu	8
19	último tempo decorrido	8
20	segundos máximos de cpu	8
21	tempo máximo decorrido	8
22	data máxima de execução (ddmmyy)	8
23	segundos mínimos de cpu	8
24	tempo mínimo decorrido	8
25	data mínima de execução (ddmmyy)	8

Nota: O tempo decorrido exibido para uma tarefa sombra é o tempo decorrido da tarefa remota ao qual ele é limitado.

Exemplos

Para extrair informações sobre o job `myjob` na estação de trabalho `main` e direcionar a saída para o arquivo `jinfo`, execute o seguinte comando:

```
jbxtract -j myjob -c main -o jinfo
```

prxtract

Extrai informações sobre prompts do banco de dados.

Sintaxe

prxtract [**-V** | **-U**] [**-o** *output*]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

-o *output*
Especifica o arquivo de saída. O padrão é **stdout**.

Resultados

Cada registro de prompt contém campos de comprimento variável, delimitados por tabulação. Os campos são descritos na Tabela 85 na página 580.

Tabela 85. Campos de Saída de Prxtract

Campo	Descrição	Comprimento Máximo (bytes)
1	nome do prompt	8
2	valor do prompt	200

Exemplos

Para extrair informações sobre todas as definições de aviso e direcionar a saída para o arquivo `prinfo`, execute o seguinte comando:

```
prxtract -o prinfo
```

caxtract

Extrai informações sobre calendários do banco de dados.

Sintaxe

caxtract [**-V** | **-U**] [**-o** *output*]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

-o *output*
Especifica o arquivo de saída. O padrão é **stdout**.

Resultados

Cada registro de calendário contém campos de comprimento variável, delimitados por tabulação. Os campos são descritos na Tabela 86.

Tabela 86. Campos de Saída de Caxtract

Campo	Descrição	Comprimento Máximo (bytes)
1	nome do calendário	8
2	descrição do calendário	64

Exemplos

Para extrair informações sobre todas as definições de calendário e direcionar a saída para o arquivo `cainfo`, execute o seguinte comando:

```
caxtract -o cainfo
```

paxtract

Extrai informações sobre parâmetros globais (variáveis) do banco de dados.

Sintaxe

paxtract [**-V** | **-U**] [**-o** *output*] [**-a**]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

-o *output*
Especifica o arquivo de saída. O padrão é **stdout**.

-a Exibe todas as variáveis definidas em todas as tabelas de variáveis. Se não for especificado, apenas as variáveis definidas na tabela de variáveis padrão serão exibidas.

Resultados

Cada registro de variável contém campos de comprimento variável, delimitados por tabulação. Os campos são descritos na Tabela 87.

Tabela 87. Campos de Saída de Paxtract

Campo	Descrição	Comprimento Máximo (bytes)
1	nome da tabela	80
2	variable name	16
3	valor da variável	72

Lembre-se: Se você não especificar a opção **-a** (todos) no comando, apenas os campos 2 e 3 serão exibidos e as variáveis listadas serão aquelas contidas na tabela de variáveis padrão apenas.

Exemplos

Para extrair informações sobre todas as definições de variáveis e direcionar a saída para o arquivo `allvarinfo`, execute o seguinte comando:

```
paxtract -a -o allvarinfo
```

rextract

Extrai informações sobre recursos do banco de dados.

Sintaxe

rextract [**-V** | **-U**] [**-o** *output*]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

-o *output*
Especifica o arquivo de saída. O padrão é **stdout**.

Resultados

Cada registro de recurso contém campos de comprimento variável, delimitados por tabulação. Os campos são descritos na Tabela 88.

Tabela 88. Campos de Saída de Rextract

Campo	Descrição	Comprimento Máximo (bytes)
1	nome da estação de trabalho	8/16
2	nome do recurso	8
3	unidades de recurso totais	4
4	descrição do recurso	72

Exemplos

Para extrair informações sobre todas as definições de recurso e direcionar a saída para o arquivo `reinfo`, execute o seguinte comando:

```
rextract -o reinfo
```

r11xtr

Extrai informações sobre fluxos de tarefa do banco de dados.

Sintaxe

r11xtr [**-V** | **-U**]
[**-m** *mm*[*yyyy*]]
[**-c** *wkstat*]
[**-o** *output*]
[**-s** *jstream_name*]

Argumentos

-V Exibe a versão do programa e sai.

-U Exibe as informações de uso do programa e sai.

-m *mm*[*yy*]
Especifica o mês (*mm*) e, opcionalmente, o ano (*yy*) dos fluxos de job. O padrão é o mês e ano atuais.

-c *wkstat*
Especifica a estação de trabalho a ser relatada. O padrão é todas as estações de trabalho.

-s *jstream_name*
Especifica o nome do fluxo de tarefas no qual as tarefas são executadas. O padrão são todos os fluxos de tarefas.

-o *output*
Especifica o arquivo de saída. O padrão é **stdout**.

Resultados

A variável ***MAESTRO_OUTPUT_STYLE*** especifica o estilo de saída para nomes longos de objetos. Com valor **LONG**, campos de comprimento integral (longo) são usados para nomes de objeto. Se a variável estiver configurada para algo que não seja **LONG**, nomes longos são truncados para oito caracteres e um sinal de mais. Por exemplo: `A1234567+`.

Cada registro de fluxo de job contém campos de comprimento variável, delimitados por tabulação. Os campos são descritos na Tabela 89.

Tabela 89. Campos de Saída de R11xtr

Campo	Descrição	Comprimento Máximo (bytes)
1	nome da estação de trabalho	16
2	nome do fluxo de tarefas	16
3	data do fluxo de job (ddmmyy)	6
4	segundos estimados de cpu	6
5	sinalizador múltiplo de estação de trabalho (* significa que alguns jobs são executados em outras estações de trabalho)	1
6	número de jobs	4
7	dia da semana (Su, Mo, Tu, We, Th, Fr, Sa)	2

Exemplos

Para extrair informações sobre os fluxos de job em junho de 2004 para a estação de trabalho`main`, execute o seguinte comando:

```
rllxtr -m 0604 -c main
```

Para extrair informações sobre os fluxos de job em junho deste ano para todas as estações de trabalho e direcionar a saída para o arquivo `rllout`, execute o seguinte comando:

```
rllxtr -m 06 -o rllout
```

xrxtrct

Extrai informações sobre referências cruzadas do banco de dados.

Sintaxe

xrxtrct [**-V** | **-U**]

Argumentos

-V Exibe a versão do comando e sai.

-U Exibe informações de utilização do comando e sai.

Resultados

A variável *MAESTRO_OUTPUT_STYLE* especifica o estilo de saída para nomes longos de objetos. Com valor **LONG**, campos de comprimento integral (longo) são usados para nomes de objeto. Se a variável estiver configurada para algo que não seja **LONG**, nomes longos são truncados para oito caracteres e um sinal de mais. Por exemplo: `A1234567+`.

A saída do comando é gravada em oito arquivos: **xdep_job**, **xdep_sched**, **xfile**, **xjob**, **xprompt**, **xresources**, **xsched** e **xwhen**. Esses arquivos são gravados no diretório atualmente em funcionamento. Você deve ter direitos de gravação e execução nesse diretório para executar o comando.

Exemplos

Para extrair informações sobre todas as referências cruzadas, execute o seguinte comando:

```
xrxtrct
```

Arquivo xdep_job

O arquivo **xdep_job** contém dois tipos de registros. O primeiro contém informações sobre jobs e fluxos de job que são dependentes de um job. Cada registro de job e fluxo de job dependente contém campos de comprimento fixo, sem delimitadores. Os campos são descritos na Tabela 90.

Tabela 90. Campos de Saída de Xdep_job

Campo	Descrição	Comprimento (bytes)
1	**03**	2
2	nome da estação de trabalho	16
3	nome da tarefa	40
4	nome do fluxo de tarefas	16

Tabela 90. Campos de Saída de Xdep_job (continuação)

Campo	Descrição	Comprimento (bytes)
5	não utilizado	240
6	nome da estação de trabalho do fluxo de job dependente	16
7	nome do fluxo de job dependente	16
8	nome da estação de trabalho do job dependente	16
9	nome do job dependente	40
10	não utilizado	6
11	não utilizado	6
12	não utilizado	8
13	fim do registro (nulo)	1

O segundo tipo de registro contém informações sobre jobs e fluxos de job que são dependentes de uma dependência entre redes. Cada registro de job e fluxo de job dependente contém campos de comprimento fixo, sem delimitadores. Os campos são descritos na Tabela 91.

Tabela 91. Campos de Saída de Xdep_job (continuação)

Campo	Descrição	Comprimento (bytes)
1	**08**	2
2	nome da estação de trabalho	16
3	nome da tarefa	120
4	não utilizado	128
5	nome da estação de trabalho do fluxo de job dependente	16
6	nome do fluxo de job dependente	16
7	nome da estação de trabalho do job dependente	16
8	nome do job dependente	40
9	não utilizado	6
10	não utilizado	6
11	não utilizado	8
12	fim do registro (nulo)	1

Arquivo xdep_sched

O arquivo **xdep_sched** contém informações sobre tarefas e fluxos de tarefas que são dependentes de um fluxo de tarefa. Cada registro de tarefa ou de fluxo de tarefa dependente contém campos de comprimento fixo, sem delimitadores. Os campos são descritos na Tabela 92.

Tabela 92. Campos de Saída de Xdep_sched

Campo	Descrição	Comprimento (bytes)
1	**02**	2
2	nome da estação de trabalho	16
3	nome do fluxo de tarefas	16
4	não utilizado	248
5	nome da estação de trabalho do fluxo de job dependente	16
6	nome do fluxo de job dependente	16
7	nome da estação de trabalho do job dependente	16
8	nome do job dependente	40
9	não utilizado	6
10	não utilizado	6
11	não utilizado	8
12	fim do registro (nulo)	1

Arquivo xfile

O arquivo **xfile** contém informações sobre jobs e fluxos de job que são dependentes de um arquivo. Cada registro contém campos de comprimento fixo, sem delimitadores. Os campos são descritos na Tabela 93 na página 585.

Tabela 93. Campos de Saída de Xfile

Campo	Descrição	Comprimento (bytes)
1	**07**	2
2	nome da estação de trabalho	16
3	nome do arquivo	256
4	nome da estação de trabalho do fluxo de job dependente	16
5	nome do fluxo de job dependente	16
6	nome da estação de trabalho do job dependente	16
7	nome do job dependente	40
8	não utilizado	6
9	não utilizado	6
10	não utilizado	8
11	fim do registro (nulo)	1

Arquivo xjob

O arquivo **xjob** contém informações sobre os fluxos de job nos quais cada job aparece. Cada registro de job contém campos de comprimento fixo, sem delimitadores. Os campos são descritos na Tabela 94.

Tabela 94. Campos de Saída de Xjob

Campo	Descrição	Comprimento (bytes)
1	**04**	2
2	nome da estação de trabalho	16
3	nome da tarefa	40
4	não utilizado	248
5	nome da estação de trabalho do fluxo de Job	16
6	nome do fluxo de tarefas	16
7	não utilizado	8
8	não utilizado	8
9	não utilizado	6
10	não utilizado	6
11	não utilizado	8
12	fim do registro (nulo)	1

Arquivo xprompt

O arquivo **xprompt** contém informações sobre jobs e fluxos de job que são dependentes de um prompt. Cada registro de prompt contém campos de comprimento fixo, sem delimitadores. Os campos são descritos na Tabela 95.

Tabela 95. Campos de Saída de Xprompts

Campo	Descrição	Comprimento (bytes)
1	**05**	2
2	nome da estação de trabalho	16
3	nome ou texto do prompt	20
4	não utilizado	236
5	nome da estação de trabalho do fluxo de job dependente	16
6	nome do fluxo de job dependente	16
7	nome da estação de trabalho do job dependente	16
8	nome do job dependente	40
9	não utilizado	6
10	não utilizado	6
11	não utilizado	8
12	fim do registro (nulo)	1

Arquivo xresource

O arquivo **xresource** contém informações sobre jobs e fluxos de job que são dependentes de um recurso. Cada registro de recurso contém campos de comprimento fixo, sem delimitadores. Os campos são descritos na Tabela 96.

Tabela 96. Campos de Saída de Xresource

Campo	Descrição	Comprimento (bytes)
1	**06**	2
2	nome da estação de trabalho	16
3	nome do recurso	8
4	não utilizado	248
5	nome da estação de trabalho do fluxo de job dependente	16
6	nome do fluxo de job dependente	16
7	nome da estação de trabalho do job dependente	16
8	nome do job dependente	40
9	unidades alocadas	6
10	não utilizado	6
11	não utilizado	8
12	fim do registro (nulo)	1

Arquivo xsched

O arquivo **xsched** contém informações sobre fluxos de job. Cada registro de fluxo de job contém campos de comprimento fixo, sem delimitadores. Os campos são descritos na Tabela 97.

Tabela 97. Campos de Saída de Xsched

Campo	Descrição	Comprimento (bytes)
1	**00**	2
2	nome da estação de trabalho	16
3	nome do fluxo de tarefas	16
4	não utilizado	248
5	nome da estação de trabalho (o mesmo de 2 acima)	16
6	nome do fluxo de job (o mesmo de 3 acima)	16
7	não utilizado	8
8	não utilizado	8
9	não utilizado	6
10	não utilizado	6
11	não utilizado	8
12	fim do registro (nulo)	1

Arquivo xwhen

O arquivo **xwhen** contém informações sobre quando os fluxos de job serão executados. Cada registro de fluxo de job contém os campos de comprimento fixo, sem delimitadores, a seguir. Os campos são descritos na Tabela 98.

Tabela 98. Campos de Saída de Xwhen

Campo	Descrição	Comprimento (bytes)
1	**01**	2
2	nome da estação de trabalho	16
3	nome ou data ON/EXCEPT	8
4	sinalizador de exceção (*=EXCEPT)	1
5	não utilizado	128
6	nome da estação de trabalho	16
7	nome do fluxo de tarefas	16
8	não utilizado	8
9	não utilizado	8
10	não utilizado	6
11	núm. de deslocamento	6
12	unidade de deslocamento	8

Tabela 98. Campos de Saída de Xwhen (continuação)

Campo	Descrição	Comprimento (bytes)
13	fim do registro (nulo)	1

Executando Relatórios do Dynamic Workload Console e relatórios em lote

É possível executar os seguintes relatórios a partir do Dynamic Workload Console:

Relatório de Histórico de Execução da Tarefa
Um relatório que coleta os dados de histórico de execução da tarefa durante um intervalo de tempo especificado. Ele é útil para detectar quais tarefas foram encerradas com erro ou estavam atrasadas, bem como tarefas críticas e promovidas e o horário mais recente no qual a tarefa pode ser iniciada sem fazer a tarefa crítica perder seu prazo final. Ele também mostra tarefas que perderam seu prazo final, tarefas de longa duração e indicadores de reexecução para reexecuções.

Relatório de Estatísticas de Execução da Tarefa
UM relatório que coleta as estatísticas de execução da tarefa. É útil detectar taxas de sucesso e de erros; duração mínima, máxima e média; estatísticas de atraso e de longa duração.

Relatório de Resumo de Carga de Trabalho da Estação de Trabalho
Um relatório que mostra a carga de trabalho nas estações de trabalho especificadas. A carga de trabalho é expressa em termos de número de tarefas que são executadas nelas. É útil para ajustes de planejamento da capacidade (modelagem de carga de trabalho e ajuste de estação de trabalho).

Relatório de Tempos de Execução de Carga de Trabalho da Estação de Trabalho
Um relatório mostrando os tempos de execução e duração das tarefas nas estações de trabalho especificadas. É útil para ajustes de planejamento da capacidade (modelagem de carga de trabalho e ajuste de estação de trabalho).

Relatório Detalhes de Produção Planejada
Um relatório baseado nas informações armazenadas em um plano de experiência ou de previsão. As informações contidas nesses planos são recuperadas do banco de dados do Tivoli Workload Scheduler. Um Relatório de Detalhes de Produção Planejada pode ser executado em mecanismos distribuídos (gerenciador de domínio principal e gerenciador de domínio backup principal). Um relatório de produção real extraído de um agente tolerante a falhas pode conter informações diferentes com relação a um plano extraído de um gerenciador de domínio principal. Por exemplo, o número de tarefas e fluxos de tarefas é o mesmo, mas o status pode alterar, pois uma tarefa bem-sucedida no principal pode estar suspensa ou pronta no agente. A taxa do status de atualização é a mesma somente no agente de status integral que é executado no principal do domínio.

Relatório Detalhes Reais de Produção
Um relatório baseado nas informações armazenadas no plano atual ou em um plano arquivado. As informações contidas nesses planos são recuperadas dos arquivos Symphony. O Relatório de Detalhes de Produção Real pode ser executado em mecanismos distribuídos (gerenciador de domínio principal, gerenciador de domínio de backup, gerenciador de domínio com conector e agente tolerante a falhas com conector).

Relatório SQL Customizado

Permite criar relatórios executando suas próprias consultas SQL. Os relatórios exibirão uma tabela com o nome de coluna, conforme especificado na parte SELECT da instrução SQL. Os dados para relatório são armazenados em um banco de dados relacional DB2 e residem do lado distribuído. O Tivoli Workload Scheduler for z/OS conecta ao banco de dados por meio da interface Java Database Connectivity (JDBC). Um driver JDBC tipo 4 é usado para conectar ao DB2 para LUW versão 8.2 ou posterior remoto.

Para obter mais informações sobre definir e executar relatórios a partir do Dynamic Workload Console, consulte a seção Guia do Usuário do Console do Dynamic Workload sobre Relatório.

Alguns desses relatórios também estão disponíveis como *relatórios em lote* e podem ser executados a partir de uma linha de comandos. Para obter informações adicionais sobre como executar relatórios em lote, consulte "Executando o relatórios em lote a partir da Interface da Linha de Comandos" na página 591.

Dependendo da interface a partir da qual o relatório é executado ou do sistema operacional do mecanismo, os formatos de saída a seguir estão disponíveis:

Tabela 99. Formatos de Saída de Relatório Suportados

Nome do relatório	Formatos de saída suportados pelo Dynamic Workload Console	Formatos de saída suportados por relatórios em lote
Relatório de Histórico de Execução da Tarefa	HTML, CSV Apenas formato de tabela	HTML, CSV, PDF Apenas formato de tabela
Relatório de Estatísticas de Execução da Tarefa	HTML, CSV Formatos de tabela e gráfico	HTML, CSV, PDF Formatos de tabela e gráfico
Relatório de Resumo de Carga de Trabalho da Estação de Trabalho	HTML, CSV Formatos de tabela e gráfico	HTML, CSV, PDF Formatos de tabela e gráfico
Relatório de Tempos de Execução de Carga de Trabalho da Estação de Trabalho	HTML, CSV Formatos de tabela e gráfico	HTML, CSV, PDF Formatos de tabela e gráfico
Relatório Detalhes de Produção Planejada	XML, CSV Apenas formato de tabela	N/D
Relatório Detalhes Reais de Produção	XML, CSV Apenas formato de tabela	N/D
Relatório SQL Customizado	HTML, CSV Apenas formato de tabela	HTML, CSV, PDF Apenas formato de tabela
Relatório geral de auditoria (Para obter informações adicionais, consulte as informações sobre como acompanhar mudanças do banco de dados usando relatórios de auditoria no *Administration Guide*)	N/D	HTML, CSV, PDF Apenas formato de tabela
Relatório detalhado de auditoria (Para obter informações adicionais, consulte as informações sobre como acompanhar mudanças do banco de dados usando relatórios de auditoria no *Administration Guide*)	N/D	HTML, CSV, PDF Apenas formato de tabela

Você deve ter as autorizações apropriadas do arquivo de segurança para objetos de relatórios para executar estes relatórios (concedidas por padrão no *tws_user* nas novas instalações). Consulte o *Administration Guide* para obter informações do arquivo de segurança.

Consulte também o*Administration Guide* para aprender como configurar oDynamic Workload Console para visualizar relatórios.

Relatórios de Histórico

A tabela a seguir resume os relatórios de histórico em termos de:

- Funcionalidade
- Critérios de seleção
- Opções de conteúdo de saída

Tabela 100. Resumo dos relatórios de histórico

Nome do Relatório	Descrição	Critérios de seleção	Opções de conteúdo de saída
Histórico de execução da tarefa	Corresponde ao **Relatório 07**. Coleta dados históricos de execução da tarefa durante um intervalo de tempo. Ajuda você a localizar: • Tarefas encerradas com erro • Tarefas atrasadas • Prazos finais ultrapassados • Longa duração • Indicadores de reexecução para reexecuções • Outras informações de histórico.	• Nome da tarefa, nome do fluxo de tarefas, nome da estação de trabalho e nome da estação de trabalho (fluxo de tarefas). Cada campo pode ser especificado usando um curinga. • Status (Sucesso, Erro, Desconhecido) • Indicadores de atraso • Intervalo de execução da tarefa • Incluir/Excluir iterações de reexecução	Você pode selecionar entre os seguintes: • Hora de início real • Duração prevista • Duração real • Número da tarefa • Iniciado com atraso (atraso) • Encerrado com atraso (atraso) • Status • Início crítico mais recente • Crítico • Promovido • Longa duração • Nome da definição de tarefa • Consumo de CPU (não disponível em estações de trabalho do Windows) • Usuário de logon • Tipo de reexecução • Número de Iteração • Código de retorno A saída está na visualização de tabela.
Estatísticas de execução da tarefa	Corresponde ao **Relatório 01**. Coleta estatísticas de execução da tarefa. Ajuda você a localizar: • Taxas de êxito/erro • Tempos mínimo e máximo decorridos e tempo de CPU • Duração média • Estatísticas de atraso e longa duração **Nota:** O relatório não inclui tarefas que tenham sido enviadas com a utilização de um nome de alias.	• Nome da tarefa, nome da estação de trabalho e login do usuário. Cada campo pode ser especificado usando um curinga. • Percentual de tarefas em Êxito, Erro, Iniciado com Atraso, Encerrado com Atraso e Longa Duração. • Total de execuções e total de reexecuções	Você pode selecionar entre os seguintes: • Detalhes da tarefa: – Usuário de logon – Criador da tarefa – Descrição – Script – Informações de Recuperação • Estatísticas da tarefa: – Total de execuções (divididas em Êxito e Erro) – Total de exceções de tempo de execução (Iniciado com Atraso, Encerrado com Atraso, Longa Duração) – Duração mínima, máxima e média e tempos de CPU (somente para execuções bem-sucedidas) – Consumo de CPU (não disponível em estações de trabalho do Windows) • Formato do relatório: – Visualização de gráficos – Visualização de Tabela – Incluir índice por tarefa ou por estação de trabalho

Tabela 100. Resumo dos relatórios de histórico (continuação)

Nome do Relatório	Descrição	Critérios de seleção	Opções de conteúdo de saída
Resumo de carga de trabalho da estação de trabalho	Fornece dados sobre a carga de trabalho em termos do número de tarefas executadas em cada estação de trabalho. Ajuda a fazer os ajustes necessários no planejamento da capacidade (modelagem da carga de trabalho e ajuste da estação de trabalho).	• Nomes da estação de trabalho. Cada campo pode ser especificado usando um curinga. • Intervalos de datas ou dias específicos para filtragem de cargas de trabalho. • Intervalos de tempo relativos (permite a reutilização da mesma tarefa de relatório para execução diária e obtenção do relatório da produção do dia anterior)	Você pode selecionar entre os seguintes: • Granularidade de informações da estação de trabalho organizada por: – Horas – Dia – Dia de produção • Opções de agregação de informações: – Forneça informações de resumo da estação de trabalho agregadas ou acumulativas para todas as estações de trabalho ou um subconjunto de estações de trabalho • Formato do relatório: – Visualização de gráficos – Visualização de Tabela – Incluir índice por data ou por estação de trabalho
Tempos de execução da carga de trabalho da estação de trabalho	Corresponde ao **Relatório 08**. Fornece dados sobre as execuções da tarefa (tempo e duração) nas estações de trabalho. Ajuda a fazer os ajustes necessários no planejamento da capacidade (modelagem da carga de trabalho e ajuste da estação de trabalho).	• Nomes da tarefa e da estação de trabalho. Cada campo pode ser especificado usando um curinga. • Período de execução da carga de trabalho • Intervalos de tempo diários	Você pode selecionar entre os seguintes: • Informações agrupadas por: – Estação de Trabalho – Data da execução Podem ser ordenadas por iteração de reexecução • Dia de produção • Informações da tarefa: – Duração real – Status – Iteração de reexecução – Nome da definição de tarefa • Formato do relatório: – Visualização de gráficos – Visualização de Tabela
SQL Customizada	Um assistente ajuda você a definir sua consulta SQL customizada (somente nas visualizações de banco de dados para as quais você possui autorização de acesso).	Os critérios especificados na consulta SQL customizada.	O relatório resultante possui uma tabela com o nome de coluna conforme especificado na parte SELECT da instrução SQL.

Relatórios de Produção

A tabela a seguir resume os relatórios de produção no que se refere a:

- Funcionalidade
- Critérios de seleção
- Opções de conteúdo de saída

Tabela 101. Resumo dos relatórios de produção

Nome do Relatório	Descrição	Critérios de seleção	Opções de conteúdo de saída
Detalhes de produção real	Corresponde ao**Relatório 10B**. Fornece dados sobre os planos atuais e arquivados.	• Nome da tarefa • Nome da estação de trabalho (tarefa) • Nome do Fluxo de Tarefas • Nome da estação de trabalho (fluxo de tarefas)	Você pode selecionar entre os seguintes: • Formato do relatório: – Simples – CSV – Microsoft Project • Incluir: – Predecessor de primeiro nível – Log da tarefa

Tabela 101. Resumo dos relatórios de produção (continuação)

Nome do Relatório	Descrição	Critérios de seleção	Opções de conteúdo de saída
Detalhes de produção planejada	Corresponde ao**Relatório 9B**. Fornece dados sobre os planos de experiência e previsão.	• Nome da tarefa • Nome da estação de trabalho (tarefa) • Nome do Fluxo de Tarefas • Nome da estação de trabalho (fluxo de tarefas)	Você pode selecionar entre os seguintes: • Formato do relatório: – Simples – CSV – Microsoft Project • Incluir: – Predecessor de primeiro nível – Log da tarefa

Executando o relatórios em lote a partir da Interface da Linha de Comandos

Esta seção descreve como é possível executar a partir da linha de comandos os relatórios listados em "Relatórios de Histórico" na página 589.

Usando uma interface da linha de comandos é possível planejar a execução desses relatórios periodicamente.

Um Cenário de Negócios de Amostra

Para evitar uma desaceleração inesperada no processamento de carga de trabalho, o analista de uma empresa grande precisa de relatórios semanais que coletem informações de histórico sobre a carga de trabalho processada para determinar e analisar quaisquer picos de carga de trabalho que possam ocorrer.

Para satisfazer essa solicitação, o TWSWEBUIDeveloper cria *Workload Workstation Summary Reports* (WWS) e *Workload Workstation Runtimes Reports* (WWR).

Para realizar essa tarefa, ele executa as seguintes etapas:

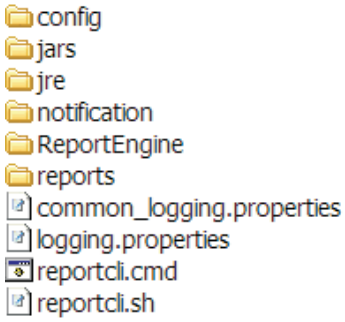
1. Ele customiza os arquivos de propriedades relacionados aos relatórios Workload Workstation Summary e Workload Workstation Runtimes, especificando o formato e o conteúdo da saída de relatório.
2. Ele planeja tarefas para obter relatórios WWS e WWR:
   - A primeira tarefa gera um relatório WWS para ser salvo localmente.
   - A segunda tarefa executa um relatório WWR durante a noite sobre períodos de tempo de picos de carga de trabalho esperados. A saída de relatório é enviada usando um e-mail para ao analista. As informações coletadas são usadas para otimizar o balanceamento de carga de trabalho nos sistemas.
3. Ele inclui as duas tarefas em um fluxo de tarefas planejado para execução semanal e gera o plano.

Configurando para Relatório de Linha de Comandos

Antes de executar relatórios em lote, você deve executar algumas etapas de configuração:

1. O software necessário para executar o relatórios em lote está contido em um pacote chamado `TWSBatchReportCli`, incluído na imagem de instalação do Tivoli Workload Scheduler, no diretório `TWSBatchReportCli`. Se planejar executar o relatórios em lote a partir de uma tarefa planejada, extraia o arquivo de pacote em um dos sistemas operacionais listados no Documento de Requisitos do Sistema em http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27041008.

   Após extrair o pacote, você terá a seguinte estrutura de diretório:



Como o utilitário tar nativo UNIX não suporta nomes de arquivos longos, se você estiver extraindo os arquivos em sistemas AIX, Solaris ou HP-UX, assegure-se de que a versão GNU mais recente de tar (gtar) esteja instalada para extrair os arquivos com êxito.

Nota:

a. Certifique de que os comandos a seguir sejam executados no diretório onde os arquivos foram extraídos:

   No UNIX

   ```
   chmod -R +x *
   chown -R nome de usuário  *
   ```

   No Windows

   Assegure que o Tivoli Workload Scheduler esteja instalado.

   ```
   setown -u username *
   ```

   em que *username* é o usuário do Tivoli Workload Scheduler que executará os relatórios.

b. Se você planeja programar tarefas que executam relatórios em lote, o sistema onde o pacote é extraído deve estar acessível como sistema de arquivos de rede a partir de um agente tolerante a falhas definido no ambiente de planejamento local.

2. Configure o arquivo de modelo `.\config\common.properties` especificando as informações para:

   a. Conecte ao banco de dados onde os dados históricos estão armazenados.

   b. Configure o formato de data e hora, incluindo o fuso horário. O arquivo `.\config\timezone.txt` contém uma lista de fuso horários suportados pelo Tivoli Workload Scheduler e as informações sobre como configurá-los. Os nomes de fuso horário fazem distinção entre maiúsculas e minúsculas.

   c. Disponibilize a saída de relatório na URL especificada no campo **ContextRootUrl**. Este é um exemplo de definições de configuração:

   ```
   ###################################################################
    # HTTP Server information
    ###################################################################

    #Specify the context root where the report will be available
    #To leverage this possibility it needs to specify in the report output dir
    #the directory that is referred by your HTTP Server with this contect root

    ContextRootUrl=http://myserver/reportoutput
   ```

   Nesse caso, certifique-se de que o *output_report_dir* especificado ao executar o comando relatórios em lote aponte para o mesmo diretório especificado no campo **ContextRootUrl**.

   d. Envie a saída de relatório usando um e-mail. Este é um exemplo de definições de configuração:

   ```
   ###################################################################
    # Email Server configuration
    ###################################################################
   ```

```
PARAM_SendReportByEmail=true

#SMTP server
mail.smtp.host=myhost.mydomain.com
#IMAP provider
mail.imap.socketFactory.fallback=false
mail.imap.port=993
mail.imap.socketFactory.port=993
#POP3 provider
mail.pop3.socketFactory.fallback=false
mail.pop3.port=995
mail.pop3.socketFactory.port=995

#####################################################################
# Email properties
#####################################################################
PARAM_EmailFrom=user1@your_company.com
PARAM_EmailTo=user2@your_company.com,user3@your_company.com
PARAM_EmailCC=user4@your_company.com
PARAM_EmailBCC=user5@your_company.com
PARAM_EmailSubject=Test send report by email
PARAM_EmailBody=This is the report attached
```

Uma explicação de todos os campos customizáveis está contida no arquivo de modelo.

Nota: Se você planeja executar relatórios do Workstation Workload Runtime, assegure que o sistema de arquivos onde o banco de dados está instalado tenha espaço livre suficiente. Se ocorrer falta de espaço em disco, uma exceção SQL como a seguinte é acionada:

```
DB2 SQL error: SQLCODE: -968, SQLSTATE: 57011
```

Executando relatórios em lote

O diretório `\reports\templates` contém um arquivo de modelo de amostra para cada tipo de relatório.

Antes de executar qualquer um desses relatórios, certifique-se de customizar o arquivo de modelo correspondente.

Nesse arquivo, denominado *report_name*`.properties`, é possível especificar:

- As informações a exibir no cabeçalho de relatório.
- Como filtrar as informações para exibir o resultado esperado.
- O formato e o conteúdo da saída de relatório.

Para obter informações sobre as configurações específicas, consulte a explicação fornecida no arquivo de modelo ao lado de cada campo.

Se estiver usando caracteres DBCS para especificar os parâmetros nos arquivos .properties de modelo, assegure-se de salvar o arquivo na codificação UTF-8.

Após configurar o ambiente como descrito em "Configurando para Relatório de Linha de Comandos" na página 591 e configurar o arquivo de modelo de relatório, use a sintaxe a seguir para executar o relatório:

reportcli -p *report_name.property*
[**-o** *output_report_dir*]
[**-r** *report_output_name*]
[**-k** key=*value*]
[**-k** key=*value*]
.....

em que:

-p *report_name.property*
Especifica o nome do caminho para o arquivo de modelo de relatório.

-o*output_report_dir*
Especifica o diretório de saída para a saída de relatório.

-r *report_output_name*
Especifica o nome da saída de relatório.

-k key=*value*
Especifica o valor das configurações. Esse valor substitui o valor correspondente, se definido, no arquivo `common.properties` ou no arquivo *report_name*`.properties`.

Exemplos

1. Neste exemplo, `reportcli.cmd` é executado com o parâmetro padrão e produz o relatório `jrh1`:

```
reportcli.cmd -p D:\ReportCLI\TWSReportCli\reports\templates\jrh.properties
-r jrh1
```

2. Neste exemplo, `reportcli.cmd` é executado usando o parâmetro `-k` para substituir os valores configurados para **PARAM_DateFormat** no arquivo `.\config\common.properties` produz o relatório `jrh1`:

```
reportcli.cmd -p D:\ReportCLI\TWSReportCli\reports\templates\jrh.properties
-r jrh2 -k PARAM_DateFormat=short
```

3. Neste exemplo, `reportcli.cmd` é executado usando o parâmetro `-k` para substituir o formato especificado para a saída de relatório no arquivo PROPERTIES produz o relatório `jrh1`:

```
./reportcli.sh -p /TWSReportCli/REPCLI/reports/templates/wwr.properties
-r wwr3 -k REPORT_OUTPUT_FORMAT=html -k OutputView=charts
```

4. Faça o seguinte se quiser executar um relatório SQL Customizado e disponibilizar a saída do relatório na URL a seguir como `http://myserver/reportoutput/report1.html`:
   a. Configure o parâmetro `ContextRootUrl` nos arquivos `common.properties` da seguinte forma:

```
######################################################################
 # HTTP Server information
 ######################################################################

 #Specify the context root where the report will be available
 #To leverage this possibility it needs to specify in the report output dir
 #the directory that is referred by your HTTP Server with this contect root

 ContextRootUrl=http://myserver/reportoutput
```

   b. Ao executar um comando relatórios em lote, especifique como *output_report_dir*, um diretório que aponte para o mesmo diretório HTTP especificado em `ContextRootUrl`. Por exemplo, se tiver mapeado localmente o `http://myserver/` como driver `R:`, é possível executar o seguinte:

```
reportclibat
   -p REPORT_CLI_DIR\reports\TWS\historical\templates\sql.properties
   -r report1
   -o R:\reportoutput
```

   c. Como uma confirmação para a execução bem-sucedida do relatório, a mensagem a seguir é exibida:

```
AWSBRC0106I Report available on: http://myserver/reportoutput/report1.html
```

      Essa URL mostra onde a saída de relatório está disponível.

Nota: Se o relatório for executado por meio de uma tarefa do Tivoli Workload Scheduler, a saída do comando é exibida na saída de tarefa.

Logs e Rastreios para relatórios em lote

O arquivo `./common_logging.properties` contém os parâmetros que é possível usar para configurar rastreio e criação de log.

O arquivo contém as configurações a seguir:

```
logFileName=reportcli.log
traceFileName=trace.log
trace=off
birt_trace=off
```

em que:

logFileName
Especifica o nome do arquivo contendo informações genéricas, avisando sobre problemas em potencial e informações sobre erros. Esse arquivo é armazenado sob `./log`.

traceFileName
Especifica o nome do arquivo que contém rastreios. Se você configurar `trace=on`, o arquivo de rastreio é armazenado sob `./log`.

rastreio
Especifica se rastreios devem ser ativados ou não. Ative os rastreios configurando `trace=on` se quiser investigar mais sobre um erro.

birt_trace
Especifica se rastreios devem ser ativados ou não para diagnosticar erros no mecanismo BIRT. Se você configurar `birt_trace=on`, um arquivo que contém o rastreio e denominado `ReportEngine_aaaa_mm_dd_hh_mm_ss.log` é armazenado na pasta `/ReportEngine/logs`

Capítulo 16. Gerenciando fusos horários

O Tivoli Workload Scheduler suporta diferentes fusos horários. Se você ativar os fusos horários, poderá gerenciar sua carga de trabalho em um ambiente de vários fusos horários.

As anotações de 3 caracteres e de comprimento variável são suportadas.

A notação de 3 caracteres é suportada para compatibilidade com versões anteriores do Tivoli Workload Scheduler.

O formato da notação de tamanho variável é área/cidade, por exemplo, Europa/Paris como equivalente a ECT (Hora Central Européia).

O capítulo é composto das seguintes seções:

- "Ativando o Gerenciamento de Fuso Horário"
- "Como o Tivoli Workload Scheduler Gerencia Fusos Horários" na página 598
- "Mudando para o Horário de Verão Ativado" na página 600
- "Mudando para o Horário de Verão Desativado" na página 600
- "Regras Gerais" na página 600

Ativando o Gerenciamento de Fuso Horário

É possível ativar ou desativar o gerenciamento de fuso horários modificando a configuração designada à opção global ***enTimeZone*** no gerenciador de domínio principal utilizando a linha de comandos **optman**. A configuração é efetivada depois que o **JnextPlan** seguinte é executado. Estas são as configurações disponíveis:

não Desativar o gerenciamento de fuso horário. Isso significa que os valores designados a todas as palavras-chave **timezone** nas definições são ignorados. Todas as restrições de tempo `at`, `until` e `deadline` são gerenciadas individualmente por cada agente tolerante a falhas, incluindo os gerenciadores principal e de domínio, ignorando, assim, o fuso horário do agente que está planejando a tarefa ou fluxo de tarefas. Em conseqüência, quando fusos horários diferentes estiverem envolvidos:

- Para tarefas, informações incorretas são exibidas sobre essas dependências de tempo quando observadas a partir de um agente diferente do proprietário da tarefa. Isso não afeta, entretanto, o processo de planejamento da tarefa.
- Para fluxos de tarefas, o impacto é que cada agente processa as dependências de tempo de acordo com seu próprio fuso horário, e, portanto, em horários diferentes, fazendo com que tarefas do mesmo fluxo de tarefas, mas definidas em agentes diferentes, sejam executadas em horários diferentes.

sim Ativar o gerenciamento de fuso horário. Isto significa que os valores designados às configurações de **timezone** são utilizados para calcular a hora em que as tarefas e fluxos de tarefas são executados nas estações de trabalho de destino.

Por padrão, a opção *enTimeZone* é configurada para **yes**.

© Copyright IBM Corp. 1999, 2014

Para obter detalhes adicionais sobre como usar a linha de comandos **optman** para gerenciar opções globais no gerenciador de domínio principal, consulte o *IBM Tivoli Workload Scheduler Administration Guide*.

Como o Tivoli Workload Scheduler Gerencia Fusos Horários

Quando o fuso horário está ativado, é possível utilizar configurações de fuso horário nas definições de estação de trabalho, tarefa e fluxo de tarefa.

Ao desempenhar atividades de gerenciamento de planejamento, o Tivoli Workload Scheduler converte o conjunto de valores para os fusos horários nas definições de objetos. As conversões são aplicadas nesta ordem:

1. Quando as instâncias do fluxo de tarefas são incluídas no plano pré-produção, o conjunto de fusos horários nas definições de fluxo de tarefa é convertido no fuso horário GMT e, em seguida, as dependências de fluxo externas são resolvidas.
2. Quando o plano de produção é criado ou estendido, as instâncias do fluxo de tarefas são designadas às estações de trabalho nas quais a instância é planejada para executar e o fuso horário é convertido de GMT no fuso horário configurado na definição da estação de trabalho de destino.

Este é o motivo pelo qual, se você utilizar os comandos **conman showsched** ou **conman showjobs** para ver as informações sobre as tarefas planejadas e os fluxos de tarefas, verá os valores de fuso horário expressos utilizando o fuso horário configurado na estação de trabalho na qual a tarefa ou fluxo de tarefas está planejado para executar. Com base na configuração da opção global *enLegacyStartOfDayEvaluation*, é possível decidir como o produto gerencia fusos horários durante o processamento e, precisamente:

Se você configurar o valor de *enLegacyStartOfDayEvaluation* **para** *no*

O valor designado à opção *startOfDay* no gerenciador de domínio principal não é convertido no fuso horário local configurado em cada estação de trabalho na rede. Isto significa que, se a opção *startOfDay* estiver configurada como 0600 no gerenciador de domínio principal, será 0600 no fuso horário local configurado em cada estação de trabalho na rede. Isto também significa que o dia do processamento começa no mesmo horário, mas não no mesmo momento, em todas as estações de trabalho.

Figura 28 na página 599 mostra como o início do dia, configurado como 0600 no gerenciador de domínio principal, é aplicado em diferentes fusos horários nos dois agentes tolerantes a falhas. A mesma conversão de tempo é aplicada nas três instâncias do emprego de tarefas **JS1** planejado para executar nas três máquinas e que contém uma dependência de tempo **at** no fuso horário `0745 US/Central`. O período de tempo que identifica o novo dia de processamento é esmaecido em Figura 28 na página 599.

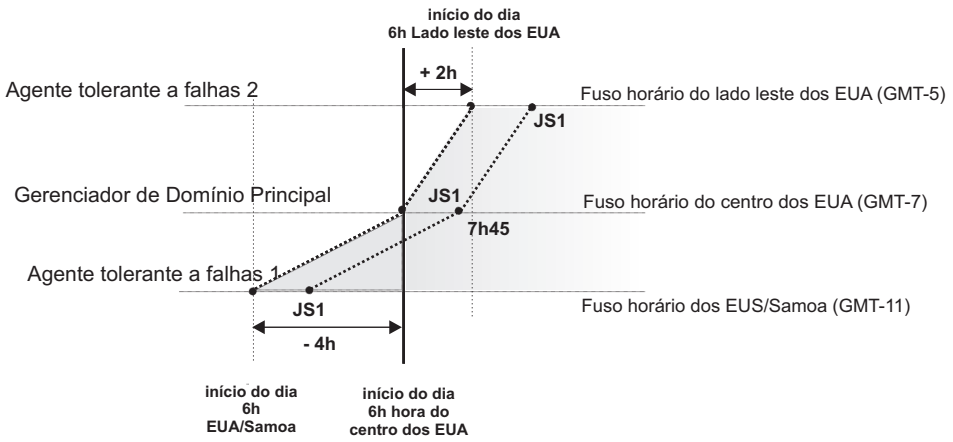


Figura 28. Exemplo quando a conversão de início do dia não é aplicada

Se você configurar o valor de *enLegacyStartOfDayEvaluation* **para** *yes***.**
O valor designado à opção *startOfDay* no gerenciador de domínio principal é convertido no fuso horário local configurado em cada estação de trabalho na rede. Isto significa que, se a opção *startOfDay* for configurada como 0600 no gerenciador de domínio principal, ela será convertida em cada estação de trabalho no horário correspondente de acordo com o fuso horário local configurado nessa estação de trabalho. Também significa que o dia do planejamento começa no mesmo momento, mas não necessariamente na mesma hora, em todas as estações de trabalho na rede.

Figura 29 mostra como o início do dia, configurado como 0600 no gerenciador de domínio principal, é aplicado em diferentes fusos horários nos dois agentes tolerantes a falhas. Também mostra como a cronometragem das três instâncias de fluxo de tarefas **JS1** planejadas para executar nas três máquinas e que contém uma dependência de tempo **at** no fuso horário `0745 US/Central` não é modificada devido à conversão *startOfDay*. O período de tempo que identifica o novo dia de processamento é esmaecido em Figura 29.

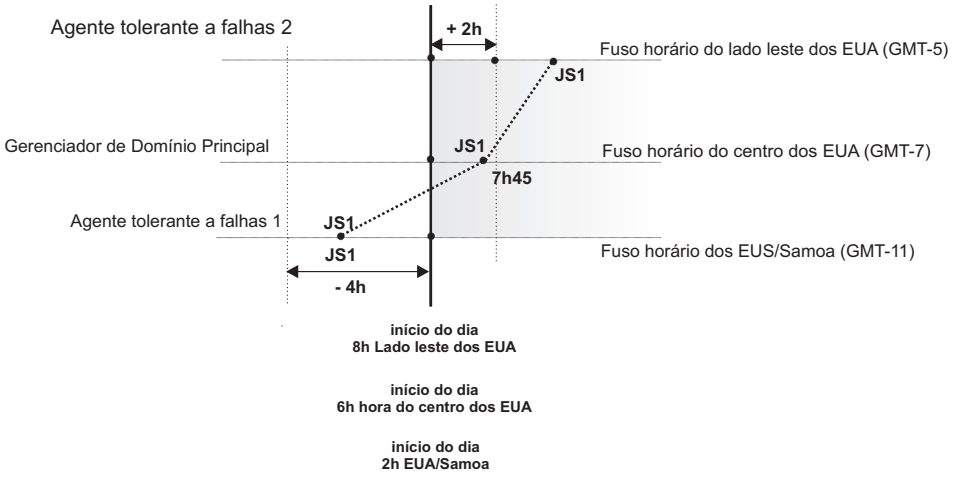


Figura 29. Exemplo quando a conversão de início do dia é aplicada

Nota: A partir da versão 8.3, não há nenhuma vinculação entre o horário configurado para *startOfDay* e o momento em que **JnextPlan** é executado.

JnextPlan pode ser executado a qualquer momento e *startOfDay* indica somente o momento em que o novo dia de processamento é iniciado.

Por padrão, a opção global *enLegacyStartOfDayEvaluation* é configurada para **no**.

Para obter detalhes adicionais sobre como usar a linha de comandos **optman** para gerenciar opções globais no gerenciador de domínio principal, consulte o *IBM Tivoli Workload Scheduler Administration Guide*.

Mudando para o Horário de Verão Ativado

Tivoli Workload Scheduler gerencia a mudança para o Horário de Verão (DST) ao gerar o plano de produção. Isto significa que a data e hora para execução designadas às tarefas e fluxos de tarefas no plano já foram convertidas na data e hora correspondentes ao DST ativado.

O exemplo a seguir explica como o Tivoli Workload Scheduler aplica a conversão de tempo quando o **JnextPlan** é executado para gerar ou estender o plano de produção enquanto a hora muda para DST.

Se o DST for ligado às 15h00, todos os fluxos de tarefas planejados para iniciar entre 14h00 e 14h59 são configurados para iniciar uma hora mais tarde. Isso acontece por causa da fase planejada de Tivoli Workload Scheduler. Por exemplo, uma tarefa definida para executar às 02h30 da manhã será planejada para executar às 03h30 da manhã.

Mudando para o Horário de Verão Desativado

Mudando para o Horário de Verão (DST) desativado, a hora do relógio é configurada para uma hora anterior em relação ao horário de DST. Para manter a consistência com os critérios de planejamento de produção, o Tivoli Workload Scheduler assegura que as instâncias do fluxo de tarefas planejadas para executar durante a hora anterior ao retrocesso do deslocamento de tempo sejam executadas somente uma vez. Como a conversão de tempo é aplicada ao gerar ou estender o plano de produção, a data e hora para execução designadas às tarefas e fluxos de tarefas no plano já estarão convertidas na data e hora correspondentes com o DST desativado.

Se um fluxo de tarefas ou uma tarefa executar em um fuso horário no qual o horário DST está desativado, isto é, o relógio está atrasado em uma hora, e se você definir uma dependência de horário para tais fluxos de tarefa ou tarefas em relação a outro fuso horário, pode ser que essa dependência de horário ocorra durante o segundo intervalo de tempo repetido. Nesse caso, a dependência de hora seria resolvida durante o primeiro intervalo de tempo.

O Tivoli Workload Scheduler reconhece que a dependência de hora ocorre no segundo intervalo de tempo repetido e a resolve apropriadamente.

Regras Gerais

Quando o fuso horário é ativado no ambiente do Tivoli Workload Scheduler, independentemente de qual valor está configurado para a opção *enLegacyStartOfDayEvaluation*, algumas regras gerais são aplicadas. Estas regras agora são descritas divididas por tópico:

Identificando as configurações de fuso horário padrão para tarefas e fluxos de tarefas:

Em uma definição de fluxo de tarefa, é possível configurar um fuso horário para o fluxo de tarefas inteiro e para as tarefas contidas no fluxo de tarefas. Esses fusos horários podem diferir uns dos outros. Para gerenciar todas as configurações de fuso horário possíveis, a conversão de fuso horário é feita respeitando os seguintes critérios:

- Se um fuso horário não for configurado para uma tarefa de um fluxo de tarefa, então, essa tarefa herda o fuso horário configurado na estação de trabalho onde a tarefa está planejada para execução.
- Se um fuso horário não for configurado para um fluxo de tarefas, então, o fuso horário configurado é o configurado na estação de trabalho onde está planejada a execução do fluxo de tarefas.
- Se nenhum dos fusos horários mencionados for configurado, então, o fuso horário utilizado é o configurado para o gerenciador de domínio principal.

Escolhendo o fuso horário correto para as estações de trabalho:

Para evitar inconsistências, antes de ativar o recurso de gerenciamento de fuso horário na rede do Tivoli Workload Scheduler, certifique-se de que, se um fuso horário estiver configurado na definição de estação de trabalho, ele será o mesmo que o fuso horário configurado no sistema no qual a estação de trabalho está instalada.

Configuração do fuso horário padrão para o gerenciador de domínio principal:

Se um fuso horário não estiver configurado na definição do gerenciador de domínio principal, ele herdará o fuso horário configurado no sistema no qual o gerenciador de domínio principal está instalado. Para ver qual fuso horário está configurado no gerenciador de domínio principal, você pode executar o seguinte comando:

```
conman showcpu;info
```

Utilizando o fuso horário nos agentes estendidos:

Os agentes estendidos herdam o fuso horário do gerenciador de domínio principal.

Exibindo a configuração do fuso horário na produção para uma dependência de tempo AT:

Se você utilizar comandos **conman sj** ou **ss** para exibir uma tarefa ou um fluxo de tarefa que possui uma dependência de tempo **at** com um fuso horário configurado, a hora especificada para a dependência **at** será exibida aplincando-se o fuso horário definido na estação de trabalho em que a tarefa ou fluxo de tarefa está definido para execução.

Aplicando um deslocamento em um fuso horário ao planejar um fluxo de tarefas:

Se você enviar na produção um fluxo de tarefas especificando uma dependência **at** com um deslocamento de +n dias, o Tivoli Workload Scheduler primeiro incluirá o deslocamento na data e, em seguida, converterá o fuso horário configurado na dependência **at**. Isto é importante especialmente ao fazer referência ao horário em que a mudança de Horário de Verão ocorre.

Como uma boa prática, se você ativar o gerenciamento de fuso horário, configure um fuso horário em cada estação de trabalho de sua rede do Tivoli Workload Scheduler.

Capítulo 17. Definindo Métodos de Acesso para Agentes

Os métodos de acesso são usados para estender as funções de planejamento de tarefa do Tivoli Workload Scheduler para outros sistemas e aplicativos. Eles executam no:

Agentes estendidos
Eles são estações de trabalho lógicas relacionadas a um método de acesso hospedado por uma estação de trabalho física do Tivoli Workload Scheduler (não outro agente estendido). Mais de uma estação de trabalho de agente estendido pode ser hospedada pela mesma estação de trabalho do Tivoli Workload Scheduler e usa o mesmo método de acesso. O agente estendido é executado em agentes tolerantes a falhas definidos usando uma definição da estação de trabalho padrão do Tivoli Workload Scheduler, que dá ao agente estendido um nome e identifica o método de acesso. O método de acesso é um programa que é executado pela estação de trabalho de hospedagem sempre que o Tivoli Workload Scheduler submeter uma tarefa a um sistema externo.

As tarefas são definidas para um agente estendido da mesma forma que para outras estações de trabalho do Tivoli Workload Scheduler, exceto que os atributos da tarefa são determinados pelo sistema externo ou pelo aplicativo.

As informações sobre a execução da tarefa são enviadas ao Tivoli Workload Scheduler a partir de um agente estendido usando o arquivo `stdlist` da tarefa. Um arquivo de opções do método pode especificar logins alternativos para ativar tarefas e verificar dependências de arquivos *abertos*. Para obter informações adicionais, consulte o *Tivoli Workload Scheduler: Guia do Usuário e Referência*.

Uma estação de trabalho física pode hospedar no máximo 255 agentes estendidos.

agentes dinâmicoseAgentes do Tivoli Workload Scheduler for z/OS
Eles se comunicam com sistemas externos para iniciar a tarefa e retornam o status da tarefa. Para executar métodos de acesso em aplicativos externos usando o agentes dinâmicos, você define uma tarefa de tipo **método de acesso** .

Os métodos de acesso estão disponíveis nos sistemas e aplicativos a seguir.
- Oracle E-Business Suite
- SAP R/3
- z/OS
- Métodos customizados
- unixssh
- unixrsh
- UNIX Local (apenas agentes tolerantes a falhas)

Os métodos de acesso do UNIX incluídos no Tivoli Workload Scheduler, são descritos em Métodos de acesso do UNIX.

Se estiver trabalhando com agentes dinâmicos, para obter informações sobre como definir estações de trabalho do Tivoli Workload Scheduler, consulte a seção que explica como definir estações de trabalho no banco de dados no *Tivoli Workload*

© Copyright IBM Corp. 1999, 2014

Scheduler: Guia do Usuário e Referência. Para obter informações sobre como gravar métodos de acesso, consulte a seção sobre a interface do método de acesso no *Tivoli Workload Scheduler: Guia do Usuário e Referência*.

Interface do método de acesso

A interface entre o Tivoli Workload Scheduler e um método de acesso consiste em informações transmitidas ao método da linha de comandos e de mensagens retornadas ao Tivoli Workload Scheduler no **stdout**.

Sintaxe da linha de comandos do método

O host do Tivoli Workload Scheduler executa um método de acesso utilizando a seguinte sintaxe de linha de comandos:

methodname **-t** *task options* **--** *taskstring*

em que:

methodname
Especifica o nome do arquivo do método de acesso. Todos os métodos de acesso devem ser armazenados no diretório: *TWS_home*/methods

-t *task* Especifica a tarefa a ser executada, em que *task* é uma das seguintes:

LJ Lança um job.

MJ Gerencia um job lançado anteriormente. Utilize essa opção para ressincronizar se uma tarefa **LJ** anterior tiver terminado inesperadamente.

CF Apenas agentes estendidos. Verifica a disponibilidade de um arquivo. Utilize essa opção para verificar as dependências `opens`.

GS Apenas agentes estendidos. Obtém o status de um job. Utilize essa opção para verificar as dependências `follows` do job.

Opções Especifica as opções associadas à tarefa. Consulte "Opções de Tarefas" para obter informações adicionais.

taskstring
Uma cadeia de até 255 caracteres associada à tarefa. Consulte o "Opções de Tarefas".

Opções de Tarefas

As opções de tarefas são listadas em Tabela 102. Um X significa que a opção é válida para a tarefa.

Tabela 102. Opções de Tarefas de Comandos do Método

Tarefa	-c	-n	-p	-r	-s	-d	-l	-o	-j	-q	-w	-S	Cadeia da Tarefa
LJ	X	X	X	X	X	X	X	X	X			X	*ljstring*
MJ	X	X	X	X	X	X	X	X	X				*mjstring*
CF	X	X	X							X			*cfstring*
GS	X	X	X	X		X					X		*gsstring*

-c *xagent*,*host*,*master*
Especifica os nomes do agente, o host e ogerenciador de domínio principal separados por vírgulas.

-n *nodename*
Especifica o nome do nó do computador associado ao agente, se houver. Isso é definido no campo **Nó** da definição da estação de trabalho do agente estendido.

-p *portnumber*
Especifica o número da porta TCP/IP associada ao agente, se houver. Isso é definido no campo **Endereço TCP** da definição da estação de trabalho do agente.

-r *currentrun*,*specificrun*
Especifica o número de execuções atual do Tivoli Workload Scheduler e o número de execuções específico associado à tarefa, separados por uma vírgula. Os números de execução atual e específico poderão ser diferentes se o job tiver sido transportado de uma execução anterior.

-s *jstream*
Especifica o nome do fluxo de tarefas da tarefa.

-d *scheddate*,*epoch*
Especifica a data do fluxo de job (*yymmdd*) e o período equivalente, separados por uma vírgula.

-l *user* Especifica o nome de usuário da tarefa. Isso é definido no campo **Logon** da definição de tarefa.

-o *stdlist*
Especifica o nome do caminho completo do arquivo de lista padrão da tarefa. Qualquer saída do job deverá ser gravada nesse arquivo.

-j *jobname*,*id*
Especifica o nome da tarefa e o identificador exclusivo designado pelo Tivoli Workload Scheduler, separados por uma vírgula. O nome é definido no campo **Nome da Tarefa** da definição de tarefa.

-q *qualifier*
Especifica o qualificador a ser utilizado em um comando de teste emitido pelo método junto ao arquivo.

-w *timeout*
Especifica a quantidade de tempo, em segundos, que o Tivoli Workload Scheduler aguarda para obter uma resposta de uma tarefa externa antes de enviar um sinal SIGTERM ao método de acesso. O padrão é 300.

-S *new name*
Especifica que o job é reexecutado utilizando esse nome no lugar do nome de job original. Em um script de job, é possível utilizar o comando `jobinfo` para retornar o nome do job e executar o script de forma diferente para cada iteração.

-- *ljstring*
Utilizado com a tarefa **LJ**. A cadeia do campo **Arquivo de Script** ou **Comando** da definição de tarefa.

-- *mjstring*
Utilizado com a tarefa **MJ**. As informações fornecidas ao Tivoli Workload Scheduler pelo método em uma mensagem indicando um alteração de estado de tarefa **%CJ** (consulte "Mensagens de resposta do método" na página 606 para obter detalhes adicionais sobre mensagens que indicam a alteração do estado da tarefa) após uma tarefa **LJ**. Em geral, isso identifica o job que foi lançado. Por exemplo, um métodoUNIX pode fornecer o PID

(Process Identification) da tarefa que foi ativada, que é então enviado pelo Tivoli Workload Scheduler como parte de uma tarefa **MJ**.

-- *cfstring*
Utilizado com a tarefa **CF**. Para a dependência `opens` de um arquivo, a cadeia do campo **Opens Files** da definição do fluxo de job.

-- *gsstring*
Utilizado com a tarefa **GS**. Especifica o job cujo status foi verificado. O formato é o seguinte:

followsjob[**,***jobid*]

em que:

followsjob
A cadeia da lista **Follows Sched/Job** da definição do fluxo de job.

jobid Um identificador de tarefa opcional recebido pelo Tivoli Workload Scheduler em uma resposta **%CJ** a uma tarefa **GS** anterior.

Mensagens de resposta do método

Os métodos retornam informações ao Tivoli Workload Scheduler em mensagens gravadas no **stdout**. Toda linha começando com um sinal de porcentagem (%) e terminando com uma nova linha, é interpretada como uma mensagem. As mensagens possuem o seguinte formato:

%CJ *state* [*mjstring* | *jobid*]

%JS [*cputime*]

%RC *rc*

%UT [*errormessage*]

em que:

CJ Altera o estado da tarefa.

state O estado para o qual o job é alterado. Todos os estados de tarefa do Tivoli Workload Scheduler são válidos, exceto HOLD e READY. Para a tarefa **GS**, os seguintes estados também são válidos:

ERROR
Ocorreu um erro.

EXTRN
Status desconhecido.

mjstring
Uma cadeia de até 255 caracteres que o Tivoli Workload Scheduler incluirá em qualquer tarefa **MJ** associada à tarefa.

jobid Uma cadeia de até 64 caracteres que o Tivoli Workload Scheduler incluirá em qualquer tarefa **GS** associada à tarefa.

JS [*cputime***]**
Indica a conclusão bem-sucedida de um job e fornece seu tempo de execução decorrido em segundos.

RC *rc* *rc* é um número interpretado pelo Tivoli Workload Scheduler como o

código de retorno do job de agente estendido. O código de retorno será considerado somente se uma condição de código de retorno for especificada na definição do job de agente estendido. Caso contrário, ele será ignorado e a conclusão bem-sucedida do job de agente estendido será indicada pela presença da mensagem **%JS** *[cputime]*. Da mesma forma, se o método não enviar a mensagem **%RC**, a conclusão bem-sucedida do job de agente estendido será indicada pela presença da mensagem **%JS** *[cputime]*.

UT [*errormessage***]**
Indica que a tarefa solicitada não é suportada pelo método. Exibe uma cadeia de até 255 caracteres que o Tivoli Workload Scheduler incluirá em sua mensagem de erro.

Arquivo de opções do método

Para agentes estendido, e Agente do Tivoli Workload Scheduler for z/OS é possível usar um arquivo de opções de método para especificar informações de login e outras opções.

Um arquivo de opções é um arquivo de texto localizado no diretório de métodos da instalação do Tivoli Workload Scheduler, contendo um conjunto de opções para customizar o comportamento do método de acesso. As opções devem ser gravadas uma por linha e ter o formato a seguir (sem espaços incluídos):

```
option=value
```

Todos os métodos de acesso usam dois tipos de arquivos de opções:

Agentes estendidos

Arquivo de opções globais
Um arquivo de configuração comum criado por padrão para cada método de acesso instalado, cujas configurações se aplicam a todas as estações de trabalho do agente estendido definidas para esse método. Quando o arquivo de opções globais é criado, ele contém apenas a opção **LJuser**, que representa o ID do usuário do sistema operacional usado para ativar o método de acesso. É possível customizar o arquivo de opções globais incluindo as opções adequadas para o método de acesso.

Arquivo de opções locais
Um arquivo de configuração que é específico para cada estação de trabalho de agente estendido dentro de uma instalação específica de um método de acesso. O nome desse arquivo é *XANAME_accessmethod*.opts, em que:

XANAME
É o nome da estação de trabalho de agente estendido. O valor para *XANAME* deve ser escrito em caracteres alfanuméricos maiúsculos. O Conjunto de caracteres de byte duplo (DBCS), o Conjunto de caracteres de byte único (SBCS) e o Texto bidirecional não são suportados.

accessmethod
É o nome do método de acesso.

Se você não criar um arquivo de opções locais, o arquivo de opções globais será utilizado. Cada estação de trabalho de agente estendido, exceto for z/OS, deve ter um arquivo de opções locais com suas próprias opções de configuração.

Por exemplo, se a instalação do método de acesso incluir duas estações de trabalho de agente estendido, CPU1 e CPU2, os nomes dos arquivos de opções locais serão respectivamente CPU1_*accessmethod*.opts e CPU2_*accessmethod*.opts.

O Tivoli Workload Scheduler lê o arquivo de opções, se ele existir, antes da execução de um método de acesso. Para agentes estendidos, se o arquivo de opções for modificado depois do Tivoli Workload Scheduler ser inicializado, as alterações entram em vigor apenas quando ele for interrompido e reinicializado.

Agentes do Tivoli Workload Scheduler for z/OS e agentes

Arquivo de opções globais

Um arquivo de configuração comum criado por padrão para cada método de acesso instalado, cujas configurações se aplicam a todas as estações de trabalho do agente definidas para esse método. Quando o arquivo de opções globais é criado, ele contém apenas a opção **LJuser**, que representa o ID do usuário do sistema operacional utilizado para executar o método de acesso. É possível customizar o arquivo de opções globais incluindo as opções adequadas para o método de acesso.

O nome do arquivo de opções globais é *accessmethod*.opts, em que o método de acesso é o nome do método que você está criando.

Arquivo de opções locais

Um arquivo de configuração que é específico para cada método de acesso. O nome deste arquivo é *optionsfile_accessmethod*.opts,

Em um ambiente distribuído:

- Se estiver definindo uma tarefa para executar o método de acesso usando o Dynamic Workload Console, ele é o arquivo de opções que você especificou na guia **Nova** > **Definição de tarefa** > **ERP** > **Método de Acesso** Tarefa XA.
- Se estiver definindo o método de acesso usando o **composer**, ele é o arquivo de opções que você especificou no atributo **target** da definição da tarefa.

Se você não criar um arquivo de opções locais, o arquivo de opções globais será utilizado.

Em um ambiente z/OS:

- Se estiver definindo uma tarefa para executar o método de acesso usando o Dynamic Workload Console, ele é o arquivo de opções que você especificou na guia **Nova** > **ERP** > **Método de Acesso** Tarefa XA.
- Se estiver definindo o método de acesso usando a instrução **JOBREC**, ele é o nome da estação de trabalho no qual o método de acesso é executado.

Se você não criar um arquivo de opções locais, o arquivo de opções globais será utilizado.

Se você não especificar uma opção no arquivo *options_file_accessmethod*.opts, o produto usará o valor especificado para essa opção no arquivo de opções globais. Se você não especificá-las nem no

arquivo *options_file_accessmethod*.opts nem no arquivo de opções globais, o produto emitirá uma mensagem de erro.

O arquivo de opções deve ter o mesmo nome de caminho de seu método de acesso, com uma extensão de arquivo .opts. Por exemplo, o nome do caminho do Windows de um arquivo de opções para um método denominado netmeth é

TWS_home\methods\netmeth.opts

O Tivoli Workload Scheduler lê o arquivo de opções, se ele existir, antes da execução de um método de acesso.

As opções reconhecidas pelo Tivoli Workload Scheduler são as seguintes:

LJuser=*username*
Especifica o logon a ser utilizado para as tarefas **LJ** e **MJ**. O padrão é o logon da definição de tarefa. Consulte "Tarefa LJ (Launch Job)" na página 610 e "Tarefa MJ (Manage Job)" na página 610.

CFuser=*username*
Apenas agentes estendidos. Especifica o logon a ser utilizado para a tarefa **CF**. O padrão para UNIX é **root** e para Windows é o nome do usuário da conta na qual o produto foi instalado. Consulte o awsrgcheckfiletask.dita.

GSuser=*username*
Especifica o logon a ser utilizado para as tarefas **GS**. O padrão para o UNIX é **root** e para o Windows é o nome do usuário da conta na qual o Tivoli Workload Scheduler foi instalado. Consulte Obter apenas agentes estendidos da tarefa de status (GS).

GStimeout=*seconds*
Especifica o período de tempo, em segundos, que o Tivoli Workload Scheduler aguarda por uma resposta antes de eliminar o método de acesso. O padrão é **300** segundos.

nodename=*node_name*
Especifica o nome do host ou o endereço IP, se for necessário pelo método que você está definindo. Para o método de acesso **unixssh**, o nome do host ou endereço IP para conexão com o mecanismo remoto.

PortNumber=*port_number*
Especifica o número da porta, se for necessário pelo método que você está definindo. Para o método de acesso **unixssh**, a porta para conexão com o mecanismo remoto.

Para Agentes do Tivoli Workload Scheduler for z/OS e agentes, é possível especificar o nome do nó e o número da porta no arquivo JobManager.ini.

Se você não especificá-los no arquivo *options_file_accessmethod*.opts, o produto usa o valor especificado no arquivo de opções globais. Se você não especificá-los nem no arquivo *options_file_accessmethod*.opts nem no arquivo de opções globais, o produto usa o valor especificado na sub-rotina *option_file* do arquivo JobManager.ini.

Nota: Se o host do agente estendido for um computador Windows, esses usuários devem ser definidos como objetos de usuários do Tivoli Workload Scheduler.

Executando Métodos

As subseções a seguir descrevem o intercâmbio entre oTivoli Workload Scheduler e um método de acesso.

Tarefa LJ (Launch Job)

A tarefa **LJ** orienta o método do agente estendido para lançar um job em um sistema ou aplicativo externo. Antes de executar o método, o Tivoli Workload Scheduler estabelece um ambiente de execução. O parâmetro **LJuser** é lido a partir do arquivo de opções do método para determinar a conta do usuário com a qual executar o método. Se o parâmetro não estiver presente ou se o arquivo de opções não existir, a conta do usuário especificada no campo **Logon** da definição da tarefa é usada. Além disso, as seguintes variáveis de ambiente são definidas:

HOME
O diretório inicial do usuário de login.

LOGNAME
O nome do usuário de login.

PATH No UNIX, é definido como `/bin:/usr/bin`. No Windows, é definido como `%SYSTEM%\SYSTEM32`.

TWS_PROMOTED_JOB
Configure como `YES`, quando a tarefa (uma tarefa essencial ou um de seus predecessores) for promovida.

TZ O fuso horário.

Se o método não puder ser executado, o job será colocado no estado FAIL.

Depois de um método ser executado, ele grava mensagens em seu **stdout** que indica o estado do job no sistema externo. A mensagens são resumidas na Tabela 103.

Tabela 103. Mensagens da Tarefa LJ (Launch Job)

Tarefa	Resposta do Método	Ação do Tivoli Workload Scheduler
LJ e **MJ**	**%CJ** *state* [*mjstring*]	Define o estado do job como *state*. Inclui *mjstring* em qualquer tarefa **MJ** subseqüente.
	%JS [*cputime*]	Define o estado do job como SUCC.
	Código de saída=diferente de zero	Define o estado do job como ABEND.
	%UT [*errormessage*] e Código de saída=2	Define o estado do job como ABEND e exibe *errormessage*.

Uma seqüência comum consiste em uma ou mais mensagens **%CJ** indicando alterações no estado do job e, em seguida, uma mensagem **%JS** antes do método sair para indicar que o job terminou com êxito. Se o job não for bem-sucedido, o método deverá sair sem gravar a mensagem **%JS**. Um método que não suporta a tarefa **LJ** grava uma mensagem **%UT** no **stdout** e sai com um código de saída **2**.

Tarefa MJ (Manage Job)

A tarefa **MJ** será utilizada para sincronização com uma tarefa ativada anteriormente se o Tivoli Workload Scheduler determinar que a tarefa **LJ** terminou inesperadamente. O Tivoli Workload Scheduler configura o ambiente da mesma maneira que para a tarefa **LJ** e transmite para ela *mjstring*. Consulte "Tarefa LJ (Launch Job)" para obter informações adicionais.

Se o método localizar o job especificado, ele responderá com as mesmas mensagens de uma tarefa **LJ**. Se o método não conseguir localizar a tarefa, ele sairá com um código de saída diferente de zero, fazendo com que o Tivoli Workload Scheduler coloque a tarefa no estado ABEND.

Eliminando um Job

Enquanto uma tarefa **LJ** ou **MJ** está sendo executada, o método deve detectar um sinal SIGTERM (sinal **15**). O sinal é enviado quando um operador emite um comando **kill** a partir do gerenciador de console do Tivoli Workload Scheduler. No recebimento do sinal, o método deve tentar parar (**kill**) o job e sair sem gravar uma mensagem **%JS**.

Apenas agentes estendidos da tarefa CF (Check File)

A tarefa **CF** solicita o método do agente estendido para verificar a disponibilidade de um arquivo no sistema externo. Antes de executar o método, o Tivoli Workload Scheduler estabelece um ambiente de execução. O parâmetro **CFuser** é lido a partir do arquivo de opções do método para determinar a conta do usuário com a qual executar o método. Se o parâmetro não estiver presente ou o arquivo de opções não existir, no UNIX, o usuário **root** é utilizado e, no Windows, o nome do usuário da conta na qual o Tivoli Workload Scheduler foi instalado é utilizado. Se o método não puder ser executado, a dependência `opens` do arquivo será marcada como falha, isto é, o status do arquivo será definido como **NO** e qualquer job ou fluxo de job dependente não terá permissão para execução.

Durante a execução, o método executa um comando de teste, ou equivalente, no arquivo que está utilizando o qualificador transmitido a ele na opção **-q** da linha de comandos. Se o teste do arquivo for verdadeiro, o método sairá com um código de saída zero. Se o teste do arquivo for falso, o método sairá com um código de saída diferente de zero. Isso está resumido na Tabela 104.

Tabela 104. Mensagens da Tarefa CF (Check File)

Tarefa	Resposta do Método	Ação do Tivoli Workload Scheduler
CF	Código de saída=0	Define o estado do arquivo como **SIM**.
	Código de saída=diferente de zero	Define o estado do arquivo como **NÃO**.
	%UT [*errormessage*] e Código de saída=2	Define o estado do arquivo como **NÃO**.

Um método que não suporta a tarefa **CF** grava uma mensagem **%UT** no **stdout** e sai com um código de saída **2**.

Apenas agentes estendidos da tarefa GS (Get Status)

A tarefa **GS** informa o método do agente estendido para verificar o status de uma tarefa. Isso é necessário quando outro job depende da conclusão bem-sucedida de um job externo. Antes de executar o método, o parâmetro **GSuser** é lido a partir do arquivo de opções do método para determinar a conta de usuário com a qual executar o método. Se o parâmetro não estiver presente ou o arquivo de opções não existir, no UNIX, o usuário **root** será utilizado e, no Windows, o nome de usuário da conta na qual o Tivoli Workload Scheduler foi instalado será utilizado. Se o método não puder ser executado, o job ou o fluxo de job dependente não terá permissão para execução. Se um *jobid* estiver disponível a partir de uma tarefa **GS** anterior, ele será transmitido para o método.

O método verifica o estado do job especificado e o retorna em uma mensagem **%CJ** gravada no **stdout**. Em seguida, sai com um código de saída zero. Com uma taxa definida pela opção local ***status de verificação de bm***, o método será executado novamente com uma tarefa **GS** até que um dos seguintes estados de job seja retornado:

abend O job terminou de forma anormal.

succ O job foi concluído com êxito.

cancl O job foi cancelado.

done O job terminou, mas seu êxito ou falha não é conhecido.

fail O job não pôde ser executado.

erro Ocorreu um erro no método durante a verificação do status do job.

extrn A verificação do job falhou ou o status do job não pôde ser determinado.

Observe que **GStimeout** no arquivo de opções do método especifica quanto tempo o Tivoli Workload Scheduler aguardará por uma resposta antes de eliminar o método. Consulte "Arquivo de opções do método" na página 607 para obter informações adicionais.

As respostas dos métodos estão resumidas na Tabela 105:

Tabela 105. Mensagens da Tarefa GS (Get Status)

Tarefa	Resposta do Método	Ação do Tivoli Workload Scheduler
GS	**%CJ** *state* [*jobid*]	Configura o estado da tarefa como *state* e inclui *jobid* em qualquer tarefa **GS** subseqüente.
	%UT [*errormessage*] e Código de saída=2	O estado do job fica inalterado.

Um método que não suporta a tarefa **GS** grava uma mensagem **%UT** no **stdout** e sai com um código de saída **2**.

Comando Cpuinfo Apenas para Agentes Estendidos

O comando **cpuinfo** pode ser utilizado em um método de acesso para retornar informações de uma definição de estação de trabalho. Consulte "Comando Cpuinfo Apenas para Agentes Estendidos" para obter informações completas sobre o comando.

Resolução de Problemas

Os tópicos a seguir são fornecidos para ajudá-lo a detectar e depurar problemas do agente estendido e do método de acesso.

Mensagens de erro da lista padrão de tarefas

Todas as mensagens de saída de um método de acesso, exceto aquelas iniciadas com um sinal de porcentagem (%), são gravadas no arquivo de lista padrão (**stdlist**) da tarefa. Para as tarefas **GS** e **CF** não associadas a tarefas do Tivoli Workload Scheduler, as mensagens são gravadas no arquivo de lista padrão do Tivoli Workload Scheduler. Para obter informações sobre problemas de qualquer tipo, verifique esses arquivos.

Método não executável

Se um método de acesso não puder ser executado, ocorrerá o seguinte:

- Para as tarefas **LJ** e **MJ**, o job será colocado no estado FAIL.
- Para a tarefa **CF**, a dependência de arquivo ficará sem solução e o job dependente permanecerá no estado HOLD.
- Para a tarefa **GS**, a dependência de job ficará sem solução e o job dependente permanecerá no estado HOLD.

Para obter informações adicionais, reveja os arquivos de lista padrão (**stdlist**) da tarefa e do Tivoli Workload Scheduler.

Mensagens do Console Manager Apenas para Agentes Estendidos

Essa mensagem de erro será exibida se você emitir um comando **start**, **stop**, **link** ou **unlink** para um agente estendido:

AWSBHU058E O comando emitido para a estação de trabalho: *workstation_name*, não pode ser desempenhado porque a estação de trabalho é um agente estendido, no qual o comando não é suportado.

Mensagens do Composer e do Compilador Apenas para Agentes Estendidos

As mensagens de erro a seguir são geradas quando o **composer** encontra sintaxe inválida em uma definição de estação de trabalho:

AWSDEM045E Há um erro na definição da estação de trabalho. A palavra-chave ACCESS não foi seguida por um método válido. Os métodos válidos correspondem ao nome de um arquivo no diretório *TWS_home*/methods (o arquivo necessário não está presente quando o método de acesso é definido).

AWSDEM046E Há um erro na definição da estação de trabalho. A palavra-chave ACCESS foi especificada mais de uma vez.

AWSDEM047E Há um erro na definição da estação de trabalho. A palavra-chave ACCESS não foi seguida por um método válido. Os métodos válidos correspondem ao nome de um arquivo no diretório *TWS_home*/methods (o arquivo necessário não está presente quando o método de acesso é definido).

Se um agente estendido for definido com um método de acesso, mas sem um host, a seguinte mensagem será exibida:

AWSBIA140E Para um agente estendido, você deve especificar o host e o método de acesso.

Mensagens Jobman Apenas para Agentes Estendidos

Para agentes estendidos, as mensagens de erro, aviso e informação são gravadas no arquivo **stdlist** do **jobman**.

O lançamento bem-sucedido de um job gera a seguinte mensagem:

AWSBDW019I Tarefa ativada *job_name*, *#Jrun_number* para o usuário *user_ID*.

O lançamento mal sucedido de um job gera a seguinte mensagem:

AWSBDW057E A tarefa *job_name* não foi ativada por esta razão: *error_message*

Falha em uma tarefa de verificação de arquivo gera a seguinte mensagem:

AWSBDW062E Jobman não conseguiu chamar o seguinte arquivo do método *method_name* para o agente estendido. O erro do sistema operacional é: *system_error*

Falha em uma tarefa de gerenciamento de job gera a seguinte mensagem:

AWSBDW066E Planman solicitou que jobman executasse uma tarefa que não é suportada no agente de destino. O seguinte arquivo de opções do método foi utilizado: *method_options_file*. O identificador da tarefa e o PID do monitor são os seguintes: *job*, *#Jmonitor_pid*

Quando um método envia uma mensagem para **jobman**, mas não é reconhecida, é gerada a seguinte mensagem:

AWSBDW064E Uma tarefa que jobman está monitorando retornou a seguinte mensagem irreconhecível: *incorrect_message*. O identificador da tarefa, PID do monitor e arquivo do método são os seguintes: *job_name*, *#Jmonitor_pid* utilizando *method file*.

Capítulo 18. Gerenciando Dependências da Interligação de Redes

As ***dependências de interligação de redes*** do Tivoli Workload Scheduler permitem que tarefas e fluxos de tarefas na rede local usem tarefas e fluxos de tarefas em uma rede remota como dependências *follows*. Este capítulo descreve como customizar seu ambiente para poder definir dependências da interligação de redes e como gerenciar as dependências da interligação de redes.

Este capítulo está dividido nas seções a seguir:

- "Visão Geral das Dependências da Interligação de Redes"
- "Configurando um agente de rede" na página 617
- "Definindo uma Dependência de Interligação de Redes" na página 619
- "Gerenciando Dependências da Interligação de Redes no Plano" na página 620
- "Dependências da interligação de redes em um ambiente misto" na página 623

Nota: Dependendo de suas necessidades e requisitos, é possível escolher entre dependências da interligação de redes e dependências cruzadas para estabelecer uma dependência entre uma tarefa em execução no mecanismo local e uma tarefa em execução em um mecanismo remoto do Tivoli Workload Scheduler. Consulte "Definindo Dependências" na página 22 para obter uma descrição sobre as diferenças entre esses dois tipos de dependências.

Visão Geral das Dependências da Interligação de Redes

Antes de especificar uma dependência de interligação de redes, você deve criar uma definição de estação de trabalho para o ***agente de rede***. Um agente de rede é uma estação de trabalho do Tivoli Workload Scheduler que manipula dependências follows entre sua rede local e uma rede remota do Tivoli Workload Scheduler.

Na rede local do Tivoli Workload Scheduler pode haver mais de um agente de rede, cada um representando uma rede remota específica do Tivoli Workload Scheduler na qual tarefas e fluxos de tarefas que referem-se a dependências da interligação de redes definidas localmente são definidos. Dependências da interligação de redes são designadas a tarefas e fluxos de tarefas da mesma maneira que dependências de sequências, com a exceção de que o nome do agente de rede está incluído para identificar a tarefa ou fluxo de tarefas seguido.

Um fluxo de tarefa especial denominado ***EXTERNAL*** é criado automaticamente pelo Tivoli Workload Scheduler para cada agente de rede na rede local. Contém tarefas sinalizadoras de substituição para representar cada dependência de interligação de redes.

Uma tarefa `EXTERNAL` é criada para cada dependência interede que pertence a fluxos de tarefa planejados para iniciar em diferentes dias com diferentes datas de planejamento. isso significa que uma tarefa `EXTERNAL` difere de outras por:

- Nome do arquivos de script, que identifica a tarefa ou o fluxo de tarefas remoto do qual a tarefa ou o fluxo de tarefas local depende.

© Copyright IBM Corp. 1999, 2014

- A data em que o fluxo de tarefa local que contém a dependência de interligação de redes está planejada para iniciar. Se a dependência for definida em uma tarefa no fluxo de tarefas, a data na qual o fluxo de tarefas está planejada para iniciar é levada em consideração.

A verificação da dependência da interligação de redes não é iniciada até o fluxo de tarefas corresponder sua dependência de tempo ou ser liberado.

No caso de duas tarefas pertencerem a diferentes fluxos de tarefa e referirem a mesma dependência de interligação de redes, enquanto um de seus fluxos de tarefa é liberado e a tarefa inicia, a dependência de interligação de redes é verificada e possivelmente liberada. Nesse caso, quando a segunda tarefa começa a verificar sua dependência de interligação de redes, encontra a dependência já solucionada, mas não necessariamente no dia esperado. Se quiser evitar que essa situação ocorra, você deve executar novamente a tarefa que representa a dependência de interligação de redes após ser solucionada pela primeira vez.

O Tivoli Workload Scheduler verifica o status das tarefas e fluxos de tarefas referidos na rede remota e mapeia seu status nas tarefas que representam as dependências da interligação de redes no fluxo de tarefa EXTERNAL. O status dessas tarefas e dos fluxos de tarefas é verificado em um intervalo de tempo fixo até a tarefa ou o fluxo de tarefas remoto atingir o estado SUCC, CANCL ou ERROR.

Compreendendo Como uma Dependência entre Redes É Mostrada

Esta seção descreve um cenário de amostra sobre as dependência de interligação de redes e como vincular a tarefa que representa a dependência de interligação de redes ao fluxo de tarefas no qual a dependência é definida. Suponhamos que:

- Você tenha definido um fluxo de tarefas denominadoELISCHED em execução na estação de trabalhoTWS206 que contém uma tarefa denominada ELI com uma dependência interede a partir de um fluxo de tarefa TWS207#FINAL.MAKEPLAN em execução em uma rede diferente do Tivoli Workload Scheduler.
- XA_MAST é o agente de rede definido na rede local para gerenciar dependências da interligação de redes de tarefas e fluxos de tarefas definidos nessa rede remota.

Utilize o comando **conman sj** para ver a dependência de interligação de redes configurada:

```
                                    (Est) (Est)
CPU  Schedule SchedTime Job State Pr Start Elapse RetCode Deps

TWS206#ELISCHE 0600 03/31 **** HOLD 10 (03/31)
                         ELI HOLD 10      XA-MAST::"TWS207#MYJS.JOB1"
```

onde (03/31) representa a restrição de tempo**at** configurada em TWS206#ELISCHE. A partir de (31/03) o status de TWS207#MYJS.JOB1 é verificado na rede remota para ver se a dependência de interligação de redes XA-MAST::"TWS207#MYJS.JOB1" é atendida.

Se você executar o comando:

```
%sj XA-MAST#EXTERNAL;info
```

verá a lista de tarefas que representa as dependências da interligação de redes definidas em tarefas e fluxos de tarefas em execução na rede local a partir de tarefas e fluxos de tarefas definidos na rede remota que pode ser alcançada através do agente de rede XA-MAST:

```
CPU  Schedule SchedTime Job  JobFile    Opt Job
 Aviso
XA-MAST #EXTERNAL
                           E8802332 TWS207#MYJS.JOB1
```

É possível ver os detalhes da tarefa ou do fluxo de tarefas dependendo de TWS207#MYJS.JOB1 na dependência de interligação de redes representada pela tarefa E8802332 no fluxo de tarefas EXTERNAL, executando o seguinte comando:

```
%sj @#EXTERNAL.E8802332;deps
```

A saída mostra o link entre a tarefa dependente e a dependência de interligação de redes:

```
                                        (Est) (Est)
 CPU  Schedule SchedTime Job State Pr Start Elapse RetCode Deps

XA-MAST#EXTERNAL.E8802332 Dependencies are:


 TWS206#ELISCHE 0600 03/31 **** HOLD 10 (03/31)
                           ELI HOLD 10      XA-MAST::"TWS207#MYJS.JOB1"
```

A verificação da dependência de interligação de redes não é iniciada até o fluxo de tarefas TWS206#ELISCHE corresponder sua dependência de tempo, (03/31), ou ser liberado.

Se houver outra tarefa definida em outro fluxo de tarefas na rede local que tem uma dependência de TWS2007#MYJS.JOB1 e o fluxo de tarefas local estiver planejado para iniciar no mesmo dia, 31/03/06, então, também a dependência dessa outra tarefa de TWS2007#MYJS.JOB1 será listada na tarefa E8802332 no fluxo de tarefas XA-MAST#EXTERNAL.

Configurando um agente de rede

As estações de trabalho do agente de rede são definidas como agentes estendidos e requerem uma estação de trabalho física hosting e um método de acesso. O método de acesso para os agentes de rede é denominado **netmth**.

O processo **batchman** no gerenciador de domínio principal consulta o **netmth** no agente de rede em intervalos de tempo fixos para obter o status da tarefa ou do fluxo de tarefas predecessor remoto. É possível customizar o intervalo de tempo entre duas verificações consecutivas, configurando a opção global *bm check status* no arquivo localopts no gerenciador de domínio principal. O Tivoli Workload Scheduler continua a verificar até a tarefa ou o fluxo de tarefa remoto alcançar o estado SUCC, CANCL ou ERROR.

Você deve criar um arquivo de opções chamado **netmth.opts** na estação de trabalho na qual o agente de rede é executado. Nesse arquivo, são definidos o usuário sob o qual o método de acesso é executado e o tempo a ser aguardado para obter uma resposta do método de acesso antes de encerrá-lo. Esse arquivo de opções deve ter o mesmo caminho que o método de acesso:

TWS_home/methods/netmth.opts

O conteúdo do arquivo netmth.opts tem a seguinte estrutura:

```
GSuser=login_name
GStimeout=seconds
```

em que:

login_name
É o login utilizado para executar o método. Se o host do agente de rede for um computador com o Windows, esse usuário deve ser definido no Tivoli Workload Scheduler.

segundos
É o número de segundos que o Tivoli Workload Scheduler aguarda por uma resposta antes de encerrar o método de acesso. A configuração padrão é 300 segundos. Na próxima vez em que o **batchman** precisar verificar o status da tarefa ou do fluxo de tarefas predecessor, o método de acesso será iniciado automaticamente.

As alterações nesse arquivo não têm efeito até você parar e reiniciar o Tivoli Workload Scheduler.

Uma Amostra da Definição do Agente de Rede

O exemplo a seguir mostra como definir uma estação de trabalho do agente de rede para uma rede remota, `Rede A`, que permite que a rede local, `Rede B`, utilize tarefas e fluxos de tarefa na rede remota como dependências da interligação de redes.

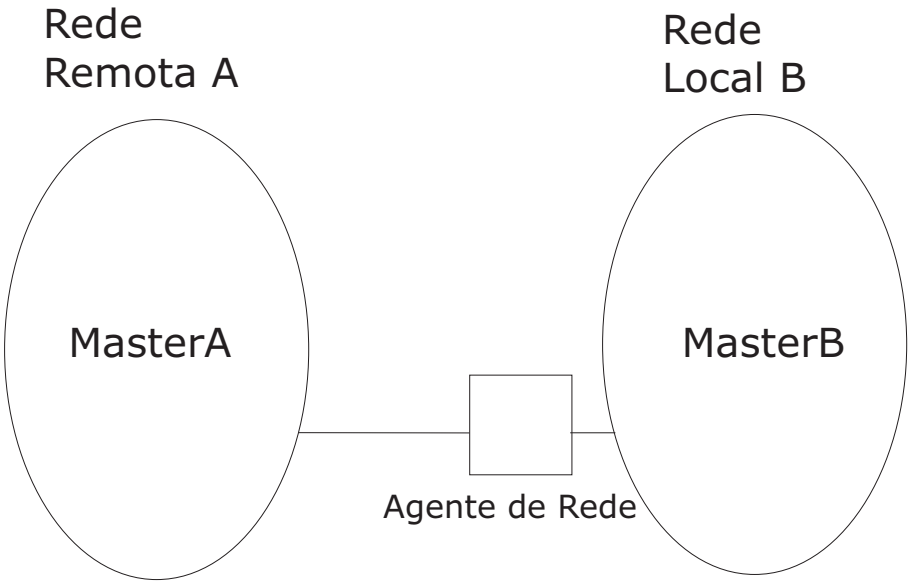


Figura 30. Redes local e remota

Suponhamos que:

- `MasterA` é ogerenciador de domínio principal da rede remota, `Network A`, e que:
  - **tws_masterA** é o*TWS_user* definido no `MasterA`.
  - O número da porta TCP para `MasterA` como 12345.
  - O nó no qual `MasterA` é definido é `MasterA.rome.tivoli.com`.
- `MasterB` é ogerenciador de domínio principal da rede local, `Rede B`, e que:
  - **tws_masterB** é o*TWS_user* definido no `MasterB`.
  - O nó no qual `MasterB` é definido é `MasterB.rome.tivoli.com`.

Uma estação de trabalho de agente de rede denominada `NetAgt`, definida no `MasterB` para gerenciar as dependências da interligação de redes de tarefas ou fluxos de tarefas definidos na `Rede A` pode ser a seguinte:

```
CPUNAME NETAGT
  DESCRIPTION "NETWORK AGENT"
  OS OTHER
  NODE MASTERA.ROME.TIVOLI.COM
  TCPADDR 12345
  FOR maestro
    HOST MASTERB
    ACCESS netmth
END
```

Importante: Grave o nome de acesso à rede `netmth` em letras minúsculas nos sistemas operacionais que fazem distinção entre maiúsculas e minúsculas.

O arquivo de opções, **`netmth.opts`** definido em `MasterB` pode ser:

```
GSuser=tws_masterB
GStimeout=600
```

Nota: O agente de rede pode ser definido no gerenciador de domínio principal ou em um agente tolerante a falhas.

Definindo uma Dependência de Interligação de Redes

A sintaxe utilizada para especificar uma dependência de interligação de redes em uma definição de fluxo de tarefa é a seguinte:

```
follows
Network_agent_name::remote_workstation#jobstreamname(time
[date]).jobname
```

em que (*time* [*date*]) são específicos do fuso horário utilizado na estação de trabalho da rede remota à qual o agente de rede está conectado; em nossa amostra, o fuso horário da `MasterA`. Se (*time* [*date*]) não for especificado nessa sintaxe ou se houver mais de um fluxo de tarefas com o mesmo (*time* [*date*]), o primeiro fluxo de tarefas é levado em consideração.

Suponhamos que:

- `schedA` seja um fluxo de tarefa definido no banco de dados `MasterA`.
- `jobA` é uma tarefa definida no banco de dados `MasterA`.
- `schedB` é um fluxo de tarefa definido no banco de dados `MasterB`.
- `jobB` é uma tarefa definida no banco de dados `MasterB`.

É possível definir dependências da interligação de redes utilizando as seguintes instruções **follows**:

Para definir uma dependência de interligação de redes `schedB` da instância do fluxo de tarefas `schedA(1100)`

Utilize a instrução a seguir:

```
schedule schedB
      on everyday
      follows NetAgt::MasterA#schedA(1100)
      :
end
```

Para definir uma dependência de interligação de redes `jobB` da `jobA` contida na instância do fluxo de tarefas `schedA(1100)`

Utilize a instrução a seguir:

```
schedule schedB
      on everyday
      :
      jobB
           follows NetAgt::MasterA#schedA(1100).jobA
end
```

Também é possível definir dependências da interligação de redes de uma tarefa em um fluxo de tarefas ou de um fluxo de tarefas em uma tarefa.

Gerenciando Dependências da Interligação de Redes no Plano

As dependências da interligação de redes são gerenciadas no plano a partir da linha de comandos **conman**, gerenciando o fluxo de tarefa EXTERNAL. No fluxo de tarefa EXTERNAL, as dependências interede são listadas como tarefas, independentemente se estão definidas para tarefas ou fluxos de tarefa. Há um fluxo de job EXTERNAL para cada agente de rede no plano.

No fluxo de tarefa EXTERNAL, nomes de tarefas exclusivos que representam as dependências da interligação de redes são gerados da seguinte forma:

```
Ennnmmss
```

em que:

nnn É um número aleatório.

mm São os minutos atuais.

ss São os segundos atuais.

O nome real da tarefa ou fluxo de tarefa é armazenado na especificação dos arquivos de script do registro de tarefa.

Nota: As tarefas e fluxos de tarefa remotos são definidos e executados em sua rede local de maneira padrão. Seu uso como dependências da interligação de redes não causa efeito em seu procedimento local.

Estados de Tarefas Definidas no Fluxo de Tarefas EXTERNAL

O status das tarefas definidas no fluxo de tarefa EXTERNAL é determinado pelo método de acesso e listado no campo Status da Liberação do fluxo de tarefa EXTERNAL. O status relatado refere-se à última vez que a rede remota foi verificada. Pode parecer que as tarefas ignoram estados quando os estados mudam entre duas verificações diferentes.

Todos os estados para tarefas e fluxos de tarefas, exceto FENCE, são listados. Além desses, há três estados que são específicos das tarefas EXTERNAL, eles são:

CANCL
A tarefa ou fluxo de tarefas correspondente na rede remota foi cancelado.

ERROR
Ocorreu um erro durante a verificação do status remoto.

EXTRN
Este é o estado inicial. Se a tarefa não for localizada na rede remota, ela permanecerá no estado EXTRN.

Nota: Se você cancelar na rede local as instâncias de tarefas ou fluxos de tarefas dependentes da mesma instância de uma tarefa ou fluxo de tarefas definido em uma rede remota, certifique-se de cancelar também manualmente a tarefa, que representa essa dependência de interligação de redes no fluxo de tarefas EXTERNAL, para evitar que o fluxo de tarefas EXTERNAL tenha continuamente o saldo transportado.A mesma consideração aplica-se quando o fluxo de tarefas local dependente da tarefa ou do fluxo de tarefas definido na rede remota não tiver o saldo transportado para o novo plano.

Trabalhando com Tarefas Definidas no Fluxo de Tarefas EXTERNAL

Essas são as ações disponíveis que podem ser executadas contra tarefas em um fluxo de tarefa EXTERNAL:

Cancelar
Cancela a tarefa EXTERNAL, liberando a dependência de interligação de redes para todas as tarefas e fluxos de tarefas locais. O status da dependência não é mais verificado.

Executar Novamente
Orienta o**conman** a reiniciar a verificação do estado do job EXTERNAL. O estado da tarefa é configurado para EXTRN imediatamente após uma ação **rerun** ser executada.

Rerun é útil para as tarefas EXTERNAL no estadoERROR. Por exemplo, se uma tarefa EXTERNAL não puder ser ativada porque o método de acesso à rede não concede permissão de execução, a tarefa entra no estadoERROR e seu status pára de ser verificado. Depois que você corrigir as permissões, o método poderá ser iniciado, mas o **conman** não iniciará a verificação do estado do job EXTERNAL até que **rerun** seja executado.

Confirm SUCC / ABEND
Configura o status da tarefa EXTERNAL para SUCC ou ABEND, liberando a dependência para todas as tarefas e fluxos de tarefa locais dependentes. O status da dependência não será mais verificado.

Nota: Nenhum desses comandos afeta a tarefa ou o fluxo de tarefas remoto na rede remota. Eles simplesmente manipulam a dependência para a rede local.

Amostra de Cenários de Gerenciamento da Dependência entre Redes

Esta seção fornece cenários de amostra que descrevem como gerenciar a dependência de interligação de redes na produção, utilizando os comandos de linha de comandos **conman**.

Supondo que você já tenha definido o seguinte:

- Uma estação de trabalho local chamada local1
- Um fluxo de tarefa definido para a estação de trabalho local local1 chamada sched1
- Uma tarefa definida em local1#sched1 chamada job1
- Um agente de rede chamadonetagt definido na rede local para gerenciar a dependência interede de tarefas e fluxos de tarefa definidos na rede remota.
- Uma estação de trabalho na rede remota chamada remote1
- Um fluxo de tarefa definido para a estação de trabalho remota remote1 chamada rcshed
- Uma tarefa definida em remote1#rsched chamada rjob

É possível executar várias ações a partir da linha de comandos **conman** na rede local. A lista a seguir contém alguns exemplos:

Incluir uma dependência interede de uma tarefa remota para uma tarefa local.
Por exemplo, para incluir a tarefa remota rjob como uma dependência de interligação de redes para job1, execute o seguinte comando:

```
adj local1#sched1.job1;follows=netagt::remote1#rsched.rjob
```

Incluir uma dependência interede de um fluxo de tarefa remoto para um fluxo de tarefa local.

Por exemplo, para incluir o fluxo de tarefas remotorsched como uma dependência de interligação de redes para o fluxo de tarefas sched1, execute o seguinte comando:

```
ads local1#sched1;follows=netagt::remote1#rsched
```

Cancelando dependências da interligação de redes gerenciadas por um agente de rede. Por exemplo, para cancelar todas as tarefas EXTERNAL para um agente de rede netagt, execute um dos dois comandos a seguir:

```
cj netagt#EXTERNAL.@
```

```
cj netagt::@
```

Confirmando a conclusão bem-sucedida de uma dependência de interligação de redes. Por exemplo, para confirmar que a tarefa remota remote1#rsched.rjob tenha sido concluída com êxito e liberar dessa forma a dependência de interligação de redes correspondente, execute o seguinte comando:

```
confirm netagt::remote1#rsched.rjob;succ
```

Excluir uma dependência de interligação de redes de uma tarefa para uma tarefa. Por exemplo, para excluir a dependência de interligação de redes da tarefa remota remote1#rsched.rjob para a tarefa local local1#sched1.job1, execute o seguinte comando:

```
ddj local1#sched1.job1;follows=netagt::remote1#rsched.rjob
```

Excluindo uma dependência de interligação de redes a partir de uma tarefa para um fluxo de tarefas.

Por exemplo, para excluir a dependência de interligação de redes da tarefa remota remote1#rsched.rjob para o fluxo de tarefa local local1#sched1, execute o seguinte comando:

```
dds local1#sched1;follows=netagt::remote1#rsched.rjob
```

Liberar uma tarefa local de uma dependência interede de uma tarefa remota.

Por exemplo, para liberar uma tarefa de uma dependência de interligação de redes de uma tarefa remota, execute o seguinte comando:

```
rj local1#sched1.job1;follows=netagt::remote1#rsched.rjob
```

Liberar um fluxo de tarefa local de uma dependência interede de uma tarefa remota.

Por exemplo, para liberar um fluxo de tarefas de uma dependência de interligação de redes de uma tarefa remota, execute o seguinte comando:

```
rs local1#sched1;follows=netagt::remote1#rsched.rjob
```

Executar novamente uma tarefa no fluxo de tarefa EXTERNAL para reiniciar a verificação de uma dependência.

Por exemplo, para executar uma tarefa novamente pertencente ao fluxo de tarefa EXTERNAL para reiniciar a verificação da dependência de interligação de redes a partir da tarefa remota remote1#rsched.rjob, execute um dos dois comandos a seguir:

```
rr netagt#EXTERNAL.rjob
```

```
rr netagt::remote1#rsched.rjob
```

Exibir dependências interede de tarefas e fluxos de tarefa definidos em uma rede remota.

Por exemplo, para exibir todas as dependências de interligação de redes definidas para um agente de rede com seus nomes originais e seus nomes de tarefas gerados, execute o seguinte comando:

```
sj netagt#EXTERNAL.@;info
```

Enviar uma tarefa com uma dependência interede de um fluxo de tarefa definido em uma rede remota

Por exemplo, para enviar um comando `rm` para o fluxo de tarefas `JOBS` com uma dependência de interligação de redes de um fluxo de tarefas remoto, execute o seguinte comando:

```
sbd "rm apfile";follows=netagt::remote1#rsched
```

Dependências da interligação de redes em um ambiente misto

Tabela 106 mostra a configuração suportada para dependências da interligação de redes definidas em um ambiente da versão 8.3 misto. A chave para a tabela é a seguinte:

Net_A O agente de rede definido na rede local.

Wks_B

A estação de trabalho na rede remota à qual o agente de rede `Net_A` está conectado. `Wks_B` é a estação de trabalho que identifica e verifica o estado da tarefa remota ou fluxo de tarefas especificado na dependência de interligação de redes.

Sym_A

O arquivo `Symphony` processado na rede local.

Sym_B

O arquivo `Symphony` processado na rede remota.

back-level

Versão 8.1, 8.2 ou 8.2.1

Tabela 106. Dependências da interligação de redes em um ambiente misto

	Net_A back-level **Sym_A back-level**	**Net_A 8.3** **Sym_A back-level**	**Net_A back-level** **Sym_A 8.3**	**Net_A 8.3** **Sym_A 8.3**
Wks_B back-level **Sym_B back-level**	Este não é um ambiente da versão 8.3 misto.	`Net_A` envia as informações para `Wks_B` como se ele tivesse a mesma versão de `Wks_B`.	`Net_A` envia as informações para `Wks_B` no formato 8.1, 8.2 ou 8.2.1. A utilização da palavra-chave *schedtime* na definição de tarefa não é suportada.	`Net_A` envia as informações para `Wks_B` como se ele tivesse a mesma versão que `Wks_B`. Se definido, a palavra-chave *schedtime* na definição de tarefa será removida automaticamente por `Net_A`.
Wks_B 8.3 **Sym_B back-level**	`Wks_B` funciona como se tivesse a mesma versão que `Net_A`.	`Net_A` envia as informações para `Wks_B`. Se definido, a palavra-chave*schedtime* na definição de tarefa será removida automaticamente por `Wks_B`.	`Net_A` envia as informações para `Wks_B`. Se definido, a palavra-chave*schedtime* na definição de tarefa será removida automaticamente por `Wks_B`.	`Net_A` envia as informações para `Wks_B`. Se definido, a palavra-chave*schedtime* na definição de tarefa será removida automaticamente por `Wks_B`.
Wks_B back-level **Sym_B 8.3**	Não suportado.	Não suportado.	Não suportado.	Não suportado.
Wks_B 8.3 **Sym_B 8.3**	Não suportado.	Não suportado.	`Net_A` envia as informações para `Wks_B`. Se definido, a palavra-chave*schedtime* é analisada por `Wks_B`.	Este é um ambiente da versão 8.3.

Capítulo 19. Definindo e Gerenciando Dependências Cruzadas

As ***dependências cruzadas*** do Tivoli Workload Scheduler ajudam a integrar e automatizar o processamento de tarefa quando:

- A carga de trabalho é distribuída por diferentes ambientes de planejamento, pois algumas das atividades são executadas em diferentes sites ou envolvem diferentes unidades organizacionais ou requerem diferentes qualificações para serem executadas.
- Mesmo se maior parte da carga de trabalho em lote for gerenciada localmente, nenhum desses ambientes está completamente isolado dos outros, pois frequentemente interoperam para trocarem ou sincronizarem dados e atividades.

Mais especificamente, o recurso de dependência cruzada é chave quando for necessário sincronizar atividades entre diferentes ambientes de planejamento de maneira fácil para que seja possível:

- Definir em um ambiente de planejamento dependências nas atividades em lote que são gerenciadas por outro ambiente de planejamento.
- Monitorar o status das tarefas predecessoras remotas como se estivessem em execução em seu ambiente local.

Além disso, é possível controlar o status dessas dependências navegando da interface com o usuário exclusiva para os diferentes ambientes de planejamento.

Este capítulo descreve como definir e usar dependências cruzadas.

Ele contém as seguintes seções:

- "Uma Introdução a Dependências Cruzadas"
- "Fluxo de Processamento no Ambiente de Planejamento Distribuído" na página 627
- "Definindo uma Dependência Cruzada" na página 629
- "Como o Status da Tarefa Sombra Muda até uma Conexão Ser Estabelecida" na página 631
- "Como uma Tarefa Sombra do z/OS É Conectada" na página 633
- "Como o Status da Tarefa Sombra Muda após a Conexão Ser Estabelecida" na página 637

Nota: Dependendo de suas necessidades e requisitos, é possível escolher entre dependências da interligação de redes e dependências cruzadas para estabelecer uma dependência entre uma tarefa em execução no mecanismo local e uma tarefa em execução em um mecanismo remoto do Tivoli Workload Scheduler. Consulte "Definindo Dependências" na página 22 para obter uma descrição das diferenças entre esses dois tipos de dependências.

Uma Introdução a Dependências Cruzadas

Uma dependência cruzada é, de um ponto de vista lógico, uma dependência que uma tarefa local tem de uma instância da tarefa que está planejada para execução em um mecanismo remoto.

Use dependências cruzadas para integrar a carga de trabalho em execução em mecanismos diferentes, que podem ser mecanismos do Tivoli Workload Scheduler

© Copyright IBM Corp. 1999, 2014

for z/OS (controlador) ou mecanismos do Tivoli Workload Scheduler (gerenciador de domínio principal e gerenciador do domínio backup principal).

Os objetos e termos a seguir são usados para descrever e implementar dependências cruzadas:

Estação de trabalho de mecanismo remoto
Um novo tipo de estação de trabalho que representa localmente um mecanismo remoto do Tivoli Workload Scheduler, distribuído ou z/OS. Esse tipo de estação de trabalho usa uma conexão baseada no protocolo HTTP ou HTTPS pare permitir que o ambiente local se comunique com o ambiente remoto.

Tarefa remota
Uma tarefa planejada para execução em um mecanismo do Tivoli Workload Scheduler remoto.

Tarefa sombra
Uma tarefa definida localmente, em uma estação de trabalho e mecanismo remoto, que é usada para mapear uma tarefa remota. A definição da tarefa sombra contém todas as informações necessárias para corresponder corretamente, no plano do mecanismo remoto, a instância da tarefa remota.

Conectar
O processo para associar uma tarefa sombra a uma instância de tarefa remota planejada no plano do mecanismo remoto do Tivoli Workload Scheduler.

De um ponto e vista lógico no ambiente local:

- A estação de trabalho do mecanismo remoto é usada para mapear o mecanismo do Tivoli Workload Scheduler remoto.
- A tarefa sombra, definida nessa estação de trabalho de mecanismo remoto, é usada para mapear uma instância da tarefa remota planejada para execução nesse mecanismo remoto do Tivoli Workload Scheduler.

Uma dependência cruzada é definida quando desejar que uma *tarefa local* (executando em seu mecanismo local) dependa de um uma *tarefa remota* (executando em um mecanismo remoto).

Para fazê-lo, você deve fazer o seguinte:

1. Crie uma tarefa *sombra* que é executada em seu mecanismo local.
2. Defina uma dependência normal que torna sua tarefa *local* dependente da tarefa *sombra*.

Quando a sombra for criada, considere que:

- Ela deve ser definida em uma estação de trabalho de tipo de mecanismo remoto, que aponta para o mecanismo remoto (ou seja, o mecanismo em que a tarefa remota está planejada para executar).
- Você deve fazê-lo apontar para a tarefa remota com a qual está criando a dependência cruzada.

A Figura 31 na página 627 mostra o fluxo lógico que implementa dependências cruzadas:

1. No processo de ligação, a tarefa sombra está associada à instância da tarefa remota.

2. Após a conexão ser estabelecida, o status da tarefa shadow é atualizado de acordo com a transição do status da tarefa remota.
3. Quando o status da tarefa sombra torna-se SUCC, a dependência normal da tarefa local é liberada e assim também a dependência cruzada da tarefa local na tarefa remota também é liberada.

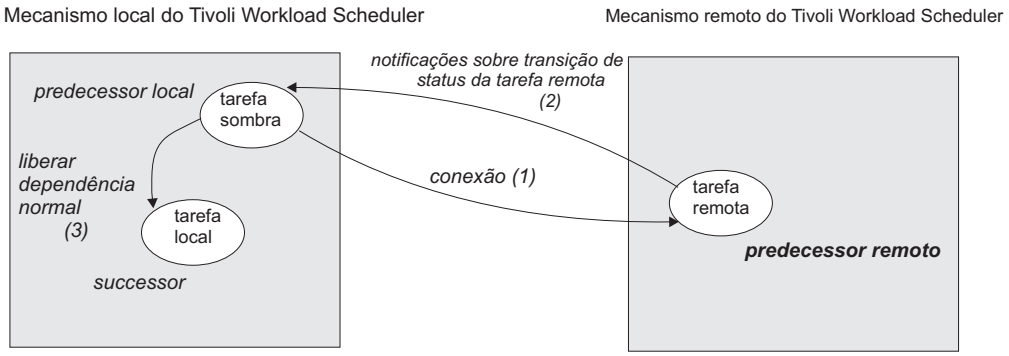


Figura 31. Lógica da Dependência Cruzada

Fluxo de Processamento no Ambiente de Planejamento Distribuído

Dependendo de se o mecanismo local emite ou recebe uma solicitação de conexão, o fluxo de processamento e os componentes envolvidos são alterados. Em ambos os casos, a estação de trabalho do broker no ambiente local deve estar ativada e em execução para permitir o gerenciamento de solicitações de conexão.

Fluxo de processamento quando o mecanismo local envia uma solicitação de conexão a um mecanismo remoto

ao definir uma tarefa sombra, você especifica as informações necessárias para estabelecer uma conexão com uma tarefa no plano de mecanismo remoto.

Quando o horário planejado da tarefa sombra chegar, se a tarefa sombra estiver livre de dependências, ela é selecionada pelo **batchman** local para envio e seu status é configurado para INTRO.

A solicitação de conexão é enviada ao mecanismo remoto. O status da tarefa sombra é configurado para WAIT.

Assim que o processamento de conexão for concluído, o mecanismo remoto envia de volta ao mecanismo local uma notificação com o resultado da conexão.

A Tabela 107 mostra como o status da tarefa sombra muda com base em:

- Se a instância a conectar existe ou não no plano do mecanismo remoto.
- O status da tarefa remota conectada.

Tabela 107. Transição de Status da Tarefa Sombra

Status da tarefa sombra no plano de produção:	Quando o mecanismo remoto:
BOUND	**z/OS** A instância do fluxo de tarefas remoto para a conexão foi localizada no plano de longo prazo ou no plano atual. **Distribuída** A instância do fluxo de tarefas remoto para a conexão foi localizada no plano de pré-produção.

Tabela 107. Transição de Status da Tarefa Sombra (continuação)

Status da tarefa sombra no plano de produção:	Quando o mecanismo remoto:
ERROR	**z/OS** Uma das situações a seguir ocorreu: • A instância do fluxo de tarefas remoto para a conexão não existe no plano de longo prazo nem no plano atual. • A instância do fluxo de tarefas remoto para a conexão foi localizada no plano de longo prazo, mas, quando é incluída no plano atual, não contém a instância de tarefa solicitada. **Distribuída** Uma das situações a seguir ocorreu: • A instância do fluxo de tarefas remoto para conexão não existe no plano de pré-produção. • A instância do fluxo de tarefas remoto para a conexão foi localizada no plano de pré-produção, mas, quando é incluída no plano de produção, não contém a instância de tarefa solicitada. • O usuário de ligação remota não é autorizado a acessar a instância da tarefa solicitada no plano de produção.
EXEC	O status da tarefa remota é EXEC.
SUCC	O status da tarefa remota é SUCC.
FAIL	O status da tarefa remota é FAIL.
ABEND	O status da tarefa remota é ABEND.
SUCC	O status da tarefa remota é CANCELED.

Nota: O status da tarefa sombra é FAIL também quando seu envio falha. Para obter detalhes adicionais sobre a transição de status da tarefa sombra, consulte "Como o Status da Tarefa Sombra Muda até uma Conexão Ser Estabelecida" na página 631 e "Como o Status da Tarefa Sombra Muda após a Conexão Ser Estabelecida" na página 637.

Fluxo de processamento quando o mecanismo remoto recebe uma solicitação de conexão do mecanismo local.

Quando o mecanismo remoto recebe uma solicitação de conexão do mecanismo local, as informações contidas na solicitação são usadas para executar a conexão no plano de pré-produção.

A solicitação de ligação também contém uma lista ordenada de URLs que o mecanismo remoto usa para enviar notificações ao mecanismo local. Se o mecanismo local for distribuído, a lista é composta da URL especificada na propriedade JDURL do arquivo chamado*TDWB_HOME*`/config/JobDispatcherConfig.properties`.

Nota: Por padrão, o Tivoli Workload Scheduler usa o TWSUser para executar a conexão no plano de produção. Se quiser limitar e controlar quais tarefas podem ser conectadas, é possível especificar um usuário diferente usando a opção global **bindUser**. O usuário especificado não precisa ser definido como um usuário no sistema operacional, ou mesmo ter uma senha, mas deve existir como entrada no arquivo de segurança com os seguintes privilégios de acesso:

- Acesso DISPLAY aos objetos *job* e *schedule* que podem ser localizados
- Acesso LIST aos objetos *job* que podem ser localizados. Esse acesso é necessário apenas se a opção global `enListSecChk` estiver configurada como `yes`.

Se os privilégios de acesso necessários não forem especificados, uma notificação com um erro é enviada de volta ao mecanismo que solicitou a

ligação.
O mecanismo remoto envia de volta o mecanismo local:

Uma notificação com o status BOUND
Se o plano de pré-produção contiver pelo menos uma instância do fluxo de tarefas especificado na solicitação de conexão e a definição desse fluxo de tarefas contiver a tarefa a conectar.

Uma notificação com o status da instância da tarefa conectada
Se a instância da tarefa a conectar for localizada no plano de produção e sempre que seu status mudar.

Uma notificação com um erro
Se a instância da tarefa para ligação não for localizada ou se o usuário da ligação não estiver autorizado.

O processo remoto do **batchman** grava uma entrada na fila do `PlanBox.msg` sempre que o status de uma tarefa remota mudar.

A cada 30 segundos, a fila do `PlanBox.msg` é verificada para novas entradas que documentem uma mudança no status das tarefas remotas que foram conectadas. Sempre que uma mudança de status for localizada, uma notificação contendo o status da tarefa remota conectada é enviado de volta ao mecanismo que solicitou a conexão.

Nota: Para alterar o intervalo de pesquisa, especifique um valor, em segundos, para **com.ibm.tws.planner.monitor.checkPlanboxInterval** no arquivo *WAS_HOME*`/profiles/`*profile_name*`/properties/TWSConfig.properties` e, em seguida, reinicie o WebSphere Application Server.

Definindo uma Dependência Cruzada

Execute estas etapas para definir uma dependência cruzada entre uma execução de tarefa em seu ambiente e outra tarefa em execução em um mecanismo diferente do Tivoli Workload Scheduler:

1. Criar uma estação de trabalho de mecanismo remoto

   Crie uma estação de trabalho e mecanismo remoto ara um mecanismo específico quando precisar definir dependências sobre instâncias de tarefas em execução nesse mecanismo remoto. Em uma estação de trabalho de mecanismo remoto, é possível executar somente tarefas sombra.

   Como uma melhor prática, se o mecanismo do Tivoli Workload Scheduler remoto for distribuído, é possível definir um conjunto dinâmico contendo uma lista ordenada de estações de trabalho do mecanismo remoto apontando para o principal remoto e para seus principais de backup, para assegurar que as capacidades de failover e do gerenciador de comutação sejam aplicadas. Para obter informações adicionais sobre o conjunto de estações de trabalho, consulte “Estação de Trabalho” na página 10.

   Nota: Recomenda-se que:

   - Todos os ambientes distribuídos envolvidos tenham o recurso de fuso horário ativado. Para obter informações adicionais, consulte a “Ativando o Gerenciamento de Fuso Horário” na página 597.
   - Você especifique como propriedade TIMEZONE das estações de trabalho de mecanismo remoto o fuso horário configurado no sistema operacional dos Gerenciadores de Domínios Principais ou Gerenciadores de Domínios Principais de Backup remotos para os quais apontam.

Para obter informações adicionais sobre as configurações específicas a serem usadas ao definir uma estação de trabalho de mecanismo remoto, consulte "Definição de estação de trabalho" na página 139.

2. Definir uma tarefa sombra na estação de trabalho do mecanismo remoto

   Crie uma tarefa sombra que aponte para uma instância de tarefa definida em um mecanismo remoto quando quiser controlar em seu ambiente local o status da tarefa remota e definir dependências cruzadas nessa tarefa remota.

   Em ambientes distribuídos do Tivoli Workload Scheduler, é possível usar alias para nomes de fluxos de tarefa e nomes de tarefas. Se você estiver definindo uma tarefa sobra definida, certifique-se de que:

   - O nome do fluxo de tarefas remoto especificado contenha o nome do fluxo de tarefas definido no banco de dados.
   - O nome da tarefa remota especificado contenha o alias, se definido, da tarefa remota a conectar.

   Se você não seguir essas diretrizes, a conexão falha e o status da tarefa sombra torna-se ERROR.

   Na definição de tarefa sombra, configure COMPLETE_IF_BIND_FAILS no campo *rccondsucc* para especificar se o status da tarefa sombra deve ser forçado para SUCC ou ERROR se a conexão no plano do mecanismo remoto falhar.

   Para obter informações adicionais sobre as configurações específicas a serem usadas ao definir uma tarefa sombra, consulte "Definição de tarefa" na página 156.

   Dependendo de se o mecanismo remoto é z/OS ou distribuído, é possível usar diferentes critérios de correspondência:

   Se o mecanismo remoto for distribuído

   É possível escolher qualquer um destes critérios de correspondência:

   Tabela 108. Critérios de Correspondência para Tarefas Sombra Distribuídas

No Dynamic Workload Console	Palavra-chave correspondente usada no composer
Precedente mais próximo	previous
Dentro de um intervalo relativo	relative from=*time_before_scheduled_time* to=*time_after_scheduled_time*
Dentro de um intervalo absoluto	absolute from=*interval_start* to=*interval_end*
Mesma data de planejamento	sameDay

   Para obter informações adicionais sobre esses critérios de correspondência, consulte "Gerenciando Dependências de Acompanhamentos Externos para Tarefas e Fluxos de Tarefas" na página 61.

   Se o mecanismo remoto for baseado no z/OS

   Anterior mais próximo é o único critério de correspondência suportado pelo Tivoli Workload Scheduler for z/OS.

   O horário planejado da instância do fluxo de tarefas que contém a tarefa sombra é usado para a correspondência no plano do mecanismo remoto.

   A transição do status da tarefa sombra é mapeado para a transição do status da instância da tarefa remota.

3. Incluir uma dependência na tarefa sombra

   É possível incluir dependência cruzada para uma tarefa local na tarefa remota definindo uma dependência para a tarefa local em uma tarefa sombra que:

   - Aponte para a instância da tarefa remota.
   - Esteja definida em uma estação de trabalho local que aponte para o mecanismo remoto no qual a tarefa remota está definida.

   A dependência cruzada na instância da tarefa remota é liberada quando a dependência local na tarefa sombra é liberada.

Monitorando uma Resolução de Dependência Cruzada no Plano de Produção

Tarefas sombra são incluídas no plano da seguinte forma:

- No tempo de execução, se uma das situações a seguir ocorrer:
  - Uma definição de tarefa sombra é enviada usando o comando **sbj**.
  - Um fluxo de tarefas contendo uma definição de tarefa sombra é enviado usando o comando **sbs**.

    Nota: Ao enviar uma tarefa sombra, especifique um fluxo de tarefas de destino com um tempo planejado conhecido para controlar melhor a instância da tarefa remota que será localizada.
- Quando o plano de pré-produção é estendido ou criado e seu período de tempo inclui o horário planejado da tarefa sombra.

Quando uma instância da tarefa sombra é incluída no plano, é possível iniciar monitoramento de seu status.

Como o Status da Tarefa Sombra Muda até uma Conexão Ser Estabelecida

A Figura 32 resume como o status de uma tarefa sombra muda até a conexão ser estabelecida.

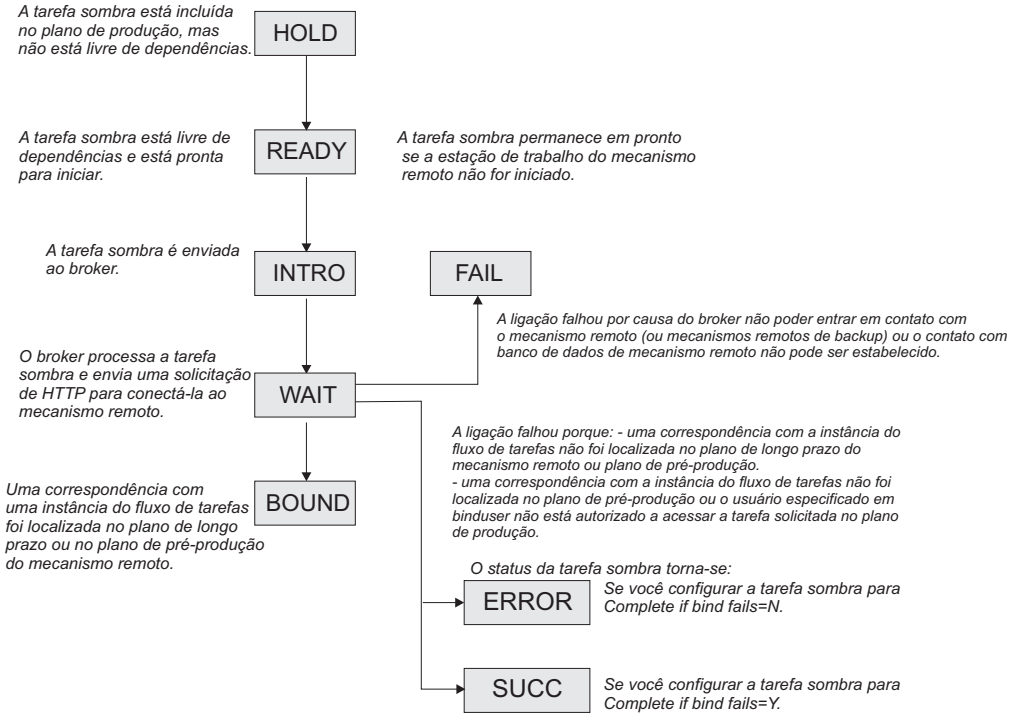


Figura 32. Transição de Status da Tarefa Sombra até a Conexão Ser Estabelecida

Como para qualquer outra tarefa, o status inicial da tarefa sombra é HOLD e muda para READY quando a tarefa for liberada de dependências e estiver pronta para iniciar.

O planejador envia então uma solicitação HTTP para o mecanismo remoto que contém as informações para identificar a tarefa sombra no plano de produção local

e as informações para identificar de forma exclusiva a instância da tarefa remota para conectar ao plano do mecanismo remoto, incluindo os critérios de correspondência. O planejador também deve ser notificado sobre o status da instância da tarefa remota conectada.

O planejador tenta entrar em contato com o mecanismo remoto, em intervalos regulares, até que um tempo limite específico expire. Se, até lá, o mecanismo remoto não puder ter sido atingido, o status da tarefa sombra é configurado para FAIL. Para alterar o tempo limite e o intervalo, especifique um valor, em segundos, tanto para MaxWaitingTime quantoStatusCheckInterval no arquivo *TDWB_HOME*/config/ResourceAdvisorConfig.properties e, em seguida, reinicie o broker.

Se o plano de pré-produção não existir no mecanismo remoto quando a solicitação de conexão for recebida, o status da tarefa sombra distribuída permanece WAIT até a geração do plano de pré-produção ser concluída e a solicitação de conexão ser processada. Isso pode ocorrer, por exemplo,quando o plano de pré-produção for criado novamente desde o início no mecanismo remoto.

Para obter informações adicionais sobre a razão de o status da tarefa sombra ser FAIL, consulte "Como Ver Por que o Status da Tarefa Sombra É FAIL" na página 638.

Quando o mecanismo remoto receber a solicitação HTTP, ele tenta identificar a instância do fluxo de tarefas a ser usado para a conexão em seu plano; o plano de pré-produção, se o mecanismo remoto for distribuído, ou o plano de longo prazo, se o mecanismo remoto for z/OS. A definição do fluxo de tarefa deve conter a definição da tarefa remota a conectar.

Para obter informações adicionais sobre como a correspondência é feita em um plano de mecanismo remoto distribuído, consulte "Como uma Tarefa Sombra Distribuída É Conectada".

Para obter informações adicionais sobre como a correspondência é feita em um plano de mecanismo remoto z/OS, consulte "Como uma Tarefa Sombra do z/OS É Conectada" na página 633.

Como uma Tarefa Sombra Distribuída É Conectada

Se o mecanismo remoto for um gerenciador de domínio principal ou um gerenciador do domínio backup principal do Tivoli Workload Scheduler, a procura pela instância da tarefa remota para conectar é feita no plano de pré-produção. Instâncias de tarefas remotas distribuídas, pertencentes aos fluxos de tarefas JOBS ou USERJOBS, não estão envolvidas no processo de conexão. Todavia, as tarefas remotas que foram movidas para USERJOBS após a ligação continuam a enviar notificações de mudança de status.

O intervalo de correspondência, exceto para os critérios de correspondência anteriores mais próximos que não requerem cálculo de intervalo, é calculado no mecanismo remoto usando as configurações especificadas na definição de tarefa sombra distribuída.

Por exemplo, quando os critérios de correspondência **sameDay** são especificados, o dia que é referido é o dia especificado no mecanismo remoto em termos de [*startOfDay*, *startOfDay*+23:59].

Ao usar um critério de correspondência baseado em intervalo, a solicitação HTTP enviada ao mecanismo remoto contém as informações a seguir para permitir que o mecanismo remoto calcule o intervalo de correspondência:

Para critérios de correspondência do intervalo absoluto:

Os valores *hhmm*, *HHMM* e, opcionalmente, *d* e *D*, especificados na cláusula:

```
<dshadow:matching>
<dshadow:absolute from="hhmm [+/-d day[s]]" to="HHMM [+/-D day[s]]"/>
</dshadow:matching>
```

Valores limites são *hhmm* -6 days e *HHMM* +6 days.

O fuso horário usado para os critérios de correspondência é o fuso horário da tarefa sombra.

Para critérios de correspondência relativos:

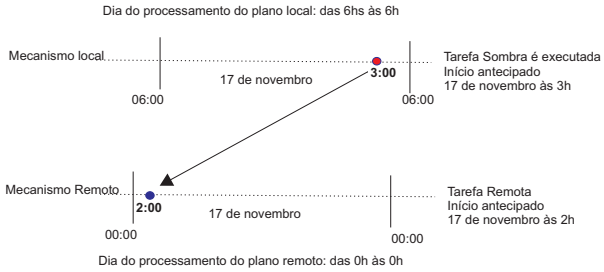
O horário planejado da tarefa sombra e os valores [*hh*]*hmm* e [*HH*]*HMM* especificados na cláusula:

```
<dshadow:matching>
<dshadow:relative from="+/-[hh]hmm" to="+/-[HH]HMM"/>
</dshadow:matching>
```

Valores limites são +/-167:59 horas.

Por exemplo, para criar uma tarefa sombra que corresponda a uma instância da tarefa remota cujo Primeiro início é 17 de novembro às 2h, é possível especificar um dos dois critérios de correspondência a seguir:

- **Mesma data planejada**
- **Dentro de um intervalo absoluto** especificando como um deslocamento: **1 dia Antes do primeiro horário de início.**



A instância da tarefa remota a corresponder é identificada no mecanismo remoto de acordo com as regras determinadas para dependências de sequências externas. Para obter detalhes adicionais sobre critérios de resolução de dependências de sequências externas, consulte “Gerenciando Dependências de Acompanhamentos Externos para Tarefas e Fluxos de Tarefas” na página 61.

Para obter informações adicionais sobre como definir tarefas sombra, consulte “Definição de tarefa” na página 156.

Como uma Tarefa Sombra do z/OS É Conectada

Se o mecanismo remoto for um controlador do IBM Tivoli Workload Scheduler for z/OS, a procura para a instância remota a conectar é feita da seguinte forma:

- Primeiro, a instância é procurada no Long Term Plan (LTP) na parte do intervalo de conexão que segue o horário de encerramento do Current Plan (CP) e precede o horário planejado da tarefa sombra.
- Se nenhuma instância for localizada, a instância é procurada no CP na parte do intervalo de conexão que precede o fim do plano atual.

Nota: Se o controlador remoto receber uma solicitação de conexão com uma URI de notificação de cliente que não está definida entre os destinos HTTP, a solicitação de conexão é descartada e a mensagem EQQHT62W é registrada em log em MLOG.

As seções a seguir descrevem os cenários que podem ocorrer ao conectar uma tarefa sombra do z/OS que tem:

- Horário planejado: 18h
- Informações de tarefa remota:
  - ID do aplicativo: **JS2**
  - Número da operação: **OP2**

Nas figuras:

- A caixa branca indica o intervalo de tempo coberto pelo LTP.
- A caixa cinza clara indica o intervalo de tempo coberto pelo CP.
- A caixa cinza escura indica o intervalo no plano do mecanismo remoto durante o qual a instância da tarefa a conectar deve ser procurada.
- A ocorrência de **JS2** destacada em negrito é a instância selecionada para a conexão.

Cenário 1: O intervalo do CP contém o horário planejado da tarefa sombra e existem ocorrências de J2S.

Figura 33mostra, destacada em negrito, a instância **JS2** que precede de

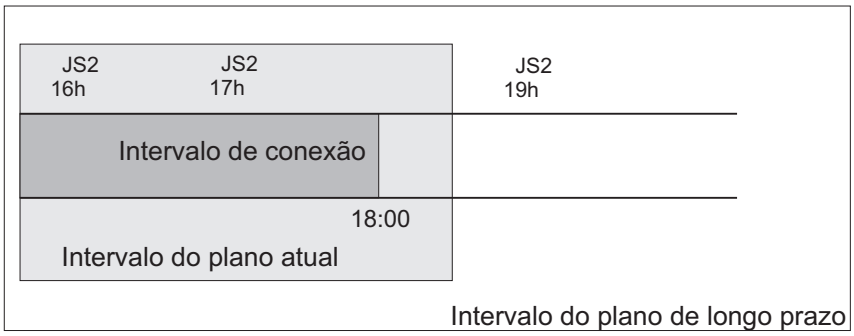


Figura 33. Instância a ser Conectada se o Horário Planejado da Tarefa Sombra Estiver Incluído no Intervalo do CP

forma mais próxima o horário planejado da tarefa sombra. Essa instância é selecionada para a conexão, pois o horário planejado está contido no CP. A tarefa sombra e a instância da tarefa remota são associadas. Se, posteriormente, uma nova instância de **JS2** que precede de forma mais próxima o horário planejado da tarefa sombra é enviada ad hoc no plano do mecanismo remoto, a correspondência com a instância **JS2** selecionada para a conexão *não* é modificada.

Neste momento, uma das situações a seguir pode ocorrer:

A instância JS2 selecionada contém OP2.

A conexão com **OP2** pertencente a **JS2** é estabelecida e uma notificação contendo:

- As informações de tarefa remota que identificam a instância **OP2** no plano de mecanismo remoto
- O status atual dessa instância **OP2**

é enviado de volta, a instância da tarefa sombra é atualizada com as informações da tarefa remota e seu status é atualizado conforme necessário.

A instância JS2 selecionada não contém mais OP2, pois foi excluído e um plano diário o remover do CP ou nunca esteve contido em J2S.

A conexão falha. Uma notificação informando que a conexão falhou é enviada de volta e o status da tarefa sombra é atualizado de acordo com o que você configurar no campo **Complete if bind fails**.

A instância JS2 selecionada contém OP2 que foi excluído, mas que ainda não foi removido do CP.

A conexão é estabelecida e uma notificação informando sobre o status de execução bem-sucedido é enviado de volta. A instância da tarefa sombra é marcada como SUCC. Suas sucessoras podem iniciar.

Cenário 2: O intervalo do plano atual contém o horário planejado da tarefa sombra, a instância JS2 que precede de forma mais próxima o horário planejado da tarefa sombra existe no LTP, mas foi cancelado no CP.

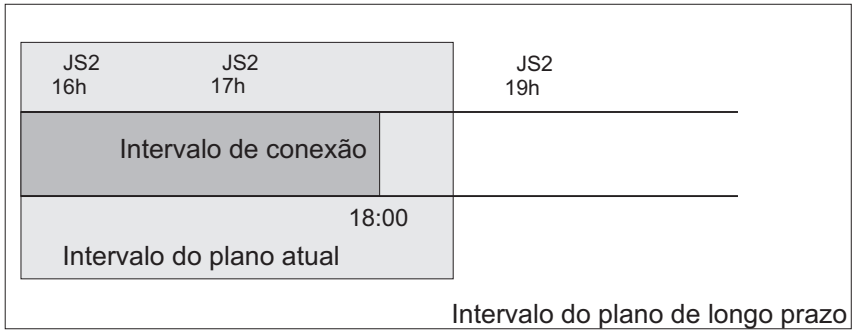


Figura 34. Instância a Ser Conectada se a Instância que Precede de Forma Mais Próxima o Horário Planejado da Tarefa Sombra Existe no LTP, mas Foi Cancelado no CP

A Figura 34 mostra, destacado em negrito, a instância **JS2** selecionada para a conexão, pois a ocorrência que melhor correspondia foi excluída.

A conexão com **OP2** pertencente a **JS2** é estabelecida e uma notificação contendo:

- As informações de tarefa remota que identificam a instância **OP2** no plano de mecanismo remoto
- O status atual dessa instância **OP2**

é enviado de volta, a instância da tarefa sombra é atualizada com as informações da tarefa remota e seu status é atualizado conforme necessário.

Cenário 3: O intervalo do CP contém o horário planejado da tarefa sombra, mas não existe nenhuma ocorrência de J2S.

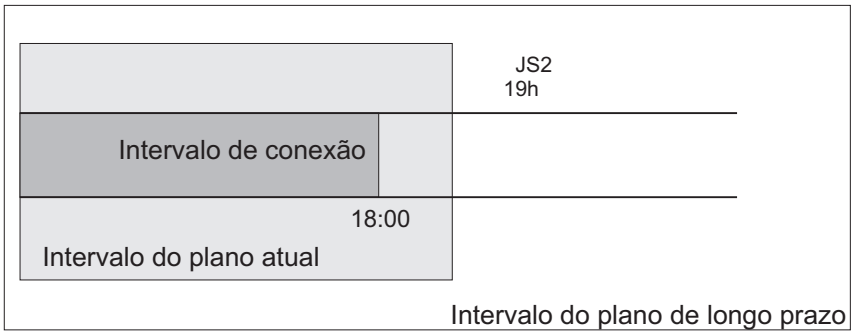


Figura 35. O Horário Planejado da Tarefa Sombra Está Incluído no CP, mas Não Existe Nenhuma Instância para Conectar

A Figura 35 mostra que uma instância **JS2** que precede de forma próxima o horário planejado da tarefa sombra não existe.

A conexão falha. Uma notificação informando que a conexão falhou é enviada de volta e o status da tarefa sombra é atualizado de acordo com o que você configurar no campo **Complete if bind fails**.

Cenário 4: O intervalo do LTP contém o horário planejado da tarefa sombra e o CP ainda não inclui a instância JS2 que precede de forma mais próxima.

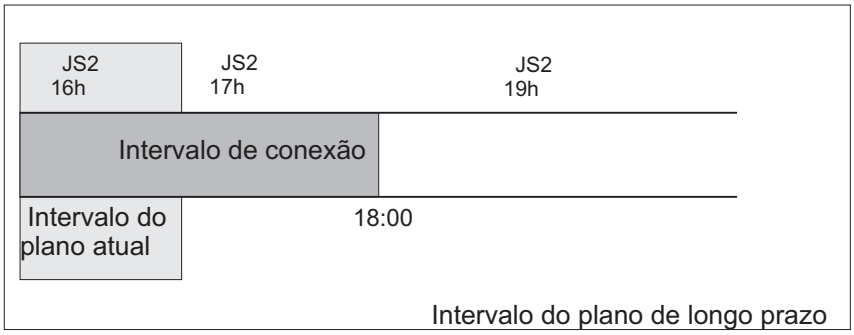


Figura 36. A Instância a Ser Conectada Existe, mas Ainda Não Está Incluída no CP

A Figura 36 mostra a instância **JS2** que pode ser associada à tarefa sombra, apesar de a tarefa **JOB2** ainda não estar no CP.

Uma notificação informando de que a conexão está estabelecida é enviada de volta e o status da tarefa sombra é configurado para BOUND.

Cenário 5: O intervalo do LTP ainda não contém o horário planejado da tarefa sombra.

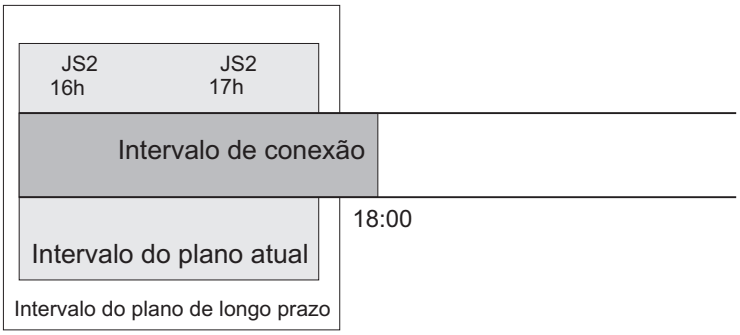


Figura 37. O Intervalo do LTP Ainda Não Contém o Horário Planejado da Tarefa Sombra

A Figura 37 mostra que nenhuma instância **JS2** pode ser associada à tarefa sombra, pois até o LTP incluir o horário planejado da tarefa sombra, instâncias **JS2** que precedem de forma mais próxima ainda podem ser incluídas.

Nesse caso, a solicitação de conexão é colocada em espera até o LTP ser estendido para incluir o horário planejado da tarefa sombra. Até lá, o status da tarefa sombra permanece WAIT.

Como o Status da Tarefa Sombra Muda após a Conexão Ser Estabelecida

Quando uma conexão é estabelecida, o mecanismo remoto envia de volta uma notificação HTTP que contém o status da conexão e, se a conexão tiver sido bem-sucedida, as informações para identificar a instância da tarefa remota conectada. Essas informações são mostradas nos detalhes da instância da tarefa sombra.

Dependendo do tipo de um mecanismo remoto, as informações a seguir sobre a instância da tarefa remota são mostradas nas propriedades da tarefa sombra:

O tipo de mecanismo remoto é distribuído

- Nome do Fluxo de Tarefas
- Hora planejada
- Estação de trabalho do fluxo de tarefas
- Nome da tarefa

O tipo de mecanismo remoto é z/OS

- ID da Aplicação
- Hora planejada
- Número da operação
- Estação de Trabalho
- Nome da tarefa, se tiver sido definido no mecanismo remoto.

Quando a instância da tarefa sombra for mapeada para uma instância da tarefa remota existente, notificações sobre mudanças de status da tarefa são enviadas de forma assíncrona do mecanismo remoto. Essas notificações são usadas para mapear a transição de status da tarefa remota para a transição de status da tarefa sombra. O mecanismo de armazenamento e encaminhamento assegura a entrega das mensagens e a recuperação em caso de falhas. A Figura 38 na página 638 mostra como o status de uma tarefa sombra distribuída muda, de quando uma conexão é

estabelecida até o status da tarefa sombra se tornar SUCC ou ERROR. Somente o status SUCC e ERROR são considerados como o status final pata uma tarefa sombra.

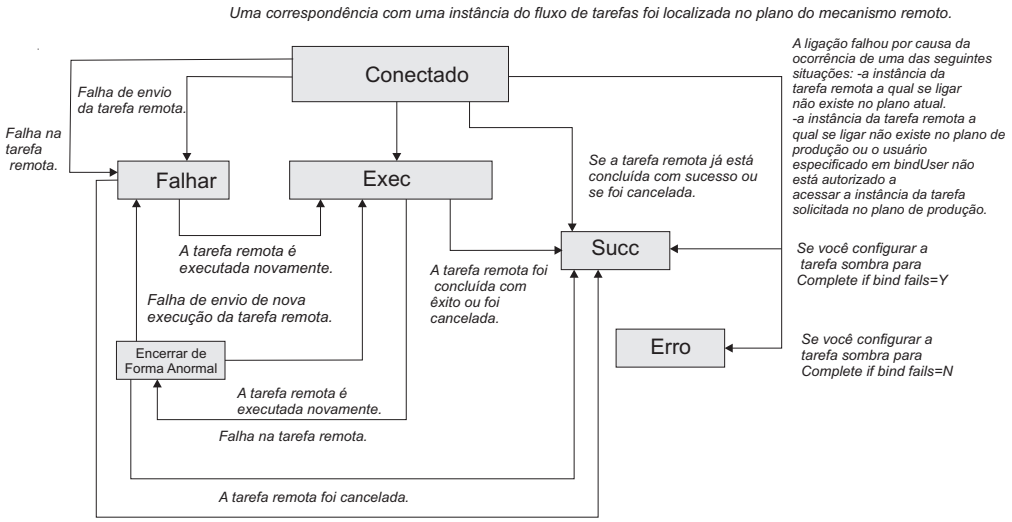


Figura 38. Cadeia de Transição de Status da Tarefa Sombra após a Conexão Ter Sido Estabelecida

Se a instância da tarefa remota já estiver concluída quando a correspondência estiver pronta, o status da tarefa sobra se torna SUCC imediatamente.

Para obter informações adicionais sobre a razão de o status da tarefa sombra ser FAIL, consulte "Como Ver Por que o Status da Tarefa Sombra É FAIL".

Quando o status da tarefa sombra satisfazer a regra de dependência, a dependência da tarefa local na tarefa sombra é resolvida e a dependência cruzada para a tarefa local na tarefa remota também é resolvida.

Como Ver Por que o Status da Tarefa Sombra É FAIL

O status da tarefa sombra pode ser FAIL em uma das seguintes situações:

- O envio da tarefa sombra falhou.
- O envio da tarefa remota falhou.

Para determinar por que o status da tarefa sombra é FAIL, consulte o log da tarefa sombra executando o comando **showjobs** com a opção `;stdlist` ou clicando em **Log da Tarefa...** para a instância da tarefa sombra na visualização **Monitorando Tarefas** no Dynamic Workload Console.

Status da Tarefa Sombra durante a Recuperação ou Nova Execução da Tarefa Remota

Após a conexão ser estabelecida, pode ocorrer que a tarefa remota conectada seja executada novamente; nesse caso, o status da tarefa sombra reflete o status da tarefa reexecutada. O status da tarefa sombra permanece EXEC enquanto a recuperação da tarefa remota está em andamento.

O status da tarefa sombra é atualizado somente quando a tarefa remota atinge um dos estados a seguir:

ABEND
Quando a tarefa remota falha em executar.

SUCC Quando a tarefa remota falha é bem-sucedida.

FAIL Quando o envio da tarefa remota falha.

É possível ver detalhes adicionais sobre a tarefa remota nas propriedades da tarefa sombra. Para ver esses detalhes:

- Execute o comando **conman showjobs** com a opção **props** com relação à tarefa sombra.
- Acesse o painel de propriedades da tarefa sombra no Dynamic Workload Console.

Como Transporte se Aplica a Dependências Cruzadas

Transporte funciona da mesma forma que as tarefas sombra fazem com outros tipos de tarefas. As tarefas sombra no status **WAIT** e **BOUND** são tratadas como tarefas no status **EXEC**. As tarefas sombra **ERROR** são tratadas como tarefas no status **FAIL** ou **ABEND**.

O status de uma tarefa sombra, conectada a uma tarefa remota que não é transportada, é configurado para ERROR quando o plano de produção remoto for estendido.

Nota: Como um melhor prática, use dependências cruzadas com fluxo de tarefas de transporte em ambos os ambientes de planejamento distribuídos local e remoto.

Para obter informações adicionais sobre a opção global **carryStates**, consulte o *Administration Guide*.

Gerenciando Tarefas Sombra no Plano de Produção

Dependendo do status da tarefa sombra, é possível executar os comandos a seguir:

Eliminar
É possível efetuar kill de uma tarefa sombra com status BOUND, EXEC ou WAIT. A associação estabelecida por meio da conexão com a tarefa remota é cancelada automaticamente e o status da tarefa sombra é configurado para ABEND com código de retorno 0.

Executar Novamente
É possível executar novamente uma tarefa sombra com status ABEND, ERROR, SUCC ou FAIL. Ao executar novamente uma tarefa sombra, uma nova solicitação de conexão é acionada.

Capítulo 20. Gerenciando um Ambiente Dinâmico do IBM i

Gerenciando agentes IBM i em um ambiente dinâmico e planejando tarefas com opções avançadas em agentes IBM i.

Definindo Agentes em Sistemas IBM i

Para iniciar o planejamento de tarefas em agentes IBM i, o agente deve estar na rede do Tivoli Workload Scheduler. No final do processo de instalação, o agente é automaticamente registrado no banco de dados do Tivoli Workload Scheduler.

É possível verificar a existência da definição da estação de trabalho do agente instalado em um sistema IBM i, usando o Dynamic Workload Console ou usando a linha de comandos do **composer**.

Para obter informações sobre como usar o console para ver as definições de estação de trabalho, consulte o *Dynamic Workload Console User's Guide*, seção sobre Editar definições de estação de trabalho.

Para obter informações sobre como usar a interface da linha de comandos para ver as definições da estação de trabalho, consulte "Definição de estação de trabalho" na página 139.

Para incluir o agente IBM i no plano, consulte *Planejamento e Instalação do Tivoli Workload Scheduler: Parte 3. Tivoli Workload Scheduler on IBM i - Configurando um Agente Dinâmico*.

Definindo Tarefas em Sistemas IBM i

Em agentes IBM i, é possível definir os seguintes tipos de tarefas com opções avançadas:

Tarefas de Serviços da Web
Para definir tarefas de serviços da web, consulte "Definição de Tarefa - Tarefas de Serviços da Web" na página 166.

Tarefas de Transferência de Arquivos
Para definir tarefas de transferência de arquivos, consulte "Definição de Tarefa - Tarefas de Transferência de Arquivos" na página 168.

tarefas do J2EE
Para definir tarefas J2EE, consulte "Definição de Tarefa - Tarefas J2EE" na página 176.

Tarefas de Banco de Dados
Para definir tarefas de banco de dados, consulte "Definição de Tarefa - Tarefas de Banco de Dados" na página 177.

tarefas de Java
Para definir tarefas Java, consulte "Definição de Tarefa - Tarefas Java" na página 181.

Tarefas Executáveis
Para definir tarefas executáveis, consulte "Definição de Tarefa - Tarefas Executáveis" na página 182.

© Copyright IBM Corp. 1999, 2014

tarefas do IBM i
Para definir tarefas do IBM i que executam comandos nativos de sistemas operacionais IBM i, consulte "Definição de Tarefa - Tarefas do IBM i" na página 187.

Tarefas de comando remoto
Para definir tarefas de comando remoto, consulte "Definição de Tarefa - Tarefas de comando remoto" na página 183.

Tarefas de Provisioning
Para definir tarefas de Provisioning, consulte "Definição de Tarefa - Tarefas de Provisioning" na página 172.

É possível definir tarefas com opções avançadas em um agente IBM i usando o Dynamic Workload Console ou usando a linha de comandos do **composer**.

Para obter mais informações sobre o procedimento para definir definições de tarefa do IBM i, consulte as seções sobre as etapas de pré-requisito para criar tipos de tarefas com opções avançadas e criar definições de tarefa no *Guia do Usuário do Tivoli Dynamic Workload Console*.

Para obter informações sobre como usar a interface da linha de comandos para criar definições de tarefa, consulte "Definição de tarefa" na página 156.

Gerenciando Agentes em Sistemas IBM i

É possível usar o Tivoli Workload Scheduler em agentes IBM i apenas para iniciar e parar os processos do agente. Para obter informações adicionais sobre como iniciar e parar agentes IBM i, consulte "Iniciando e Parando Agentes em Sistemas IBM i".

Para gerenciar o agente IBM i, use os utilitários descritos em "Usando Comandos Utilitários para Agentes em Sistemas IBM i".

Iniciando e Parando Agentes em Sistemas IBM i

É possível usar o Tivoli Workload Scheduler em agentes IBM i apenas para iniciar e parar os processos do agente.

Iniciando Agentes em Sistemas IBM i:
Use o utilitário **StartUpLwa**.

Para obter informações sobre o utilitário para iniciar agentes no IBM i, consulte "StartUpLwa" na página 528.

Parando Agentes em Sistemas IBM i:
Use o utilitário **ShutDownLwa**.

Para obter informações sobre o utilitário para parar agentes no IBM i, consulte "ShutDownLwa" na página 527.

Usando Comandos Utilitários para Agentes em Sistemas IBM i

Para agentes em sistemas IBM i, é possível usar os seguintes utilitários:

param Para usar o utilitário param, consulte "param" na página 543.

twstrace
Para usar o utilitário twstrace, consulte "twstrace" na página 557.

recurso
Para usar o utilitário resource, consulte “recurso” na página 546.

cpuinfo
Para usar o utilitário cpuinfo, consulte “cpuinfo” na página 497.

version
Para usar o utilitário version, consulte “version” na página 529.

Planejando Tarefas em Sistemas IBM i

Ao planejar uma tarefa em sistemas IBM i, a tarefa ativa um comando nativo que pode ser um comando do sistema ou de usuário. O comando nativo consiste no comando do sistema SBMJOB, que ativa uma tarefa em lote. O comando nativo inicia uma ou mais tarefas em lote. As tarefas em lote podem ser monitoradas apenas se forem iniciadas pelo comando nativo. O IBM i Agent Monitor pode monitorar um máximo de 130 tarefas em lote.

O log da tarefa do agente e a variável de ambiente TWSASPOOLS

Por padrão, todas as informações sobre a execução de tarefas são armazenadas no log da tarefa do agente. A maioria dessas informações normalmente consiste em arquivos de spool. Para selecionar os tipos de arquivos de spool que você deseja incluir no log da tarefa do agente, use a variável do sistema **TWSASPOOLS**, que funciona no nível do agente IBM i para qualquer tarefa a ser enviada.

A variável do sistema **TWSASPOOLS** força o agente IBM i a ignorar todos os arquivos de spool ou incluir um ou mais deles.

No agente IBM i, crie uma nova variável de ambiente de nível de sistema chamada TWSASPOOLS e configure-a para uma lista de tipos de arquivos de spool que devem ser incluídos. A lista deve iniciar com o token **SPOOLS**:.

Por exemplo, para forçar o agente IBM i a ignorar todos os arquivos de spool, crie a variável TWSASPOOLS da seguinte forma.

```
ADDENVVAR ENVVAR(TWSASPOOLS) VALUE(SPOOLS:) LEVEL(*SYS)
```

em que a lista após o token SPOOL: está vazia. Neste caso, qualquer relatório do log da tarefa do agente para o agente IBM i está limitado ao relatório de atividades que o Agent Monitor produz para rastrear sua ação de envio e monitoramento e ao log da tarefa do IBM i do Agent Monitor, que é sempre incluído no final do log da tarefa do agente.

Para permitir que o agente IBM i inclua apenas os tipos de arquivos de spool QPRINT e QPJOBLOG, ou seja, todos os arquivos de spool produzidos pelas instruções **printf** dentro de qualquer programa ILE-C e todos os logs de tarefa produzidos, crie a TWSASPOOLS da seguinte forma:

```
ADDENVVAR ENVVAR(TWSASPOOLS) VALUE('SPOOLS: QPRINT QPJOBLOG')
LEVEL(*SYS)
```

Se a variável TWSASPOOLS já existir, altere-a da seguinte forma:

```
CHGENVVAR ENVVAR(TWSASPOOLS)
VALUE('SPOOLS: QPRINT QPJOBLOG') LEVEL(*SYS)
```

Se qualquer parâmetro VALUE for configurado para uma sequência incorreta, o agente IBM i ignora a opção da variável de ambiente TWSASPOOLS. É possível

criar e alterar a variável de ambiente TWSASPOOLS com o agente IBM i ativo, mas nenhuma atividade de carga de trabalho deve estar em execução.

Monitoramento de Tarefas-filhas em Agentes IBM i

Ao enviar um comando em um agente IBM i, o comando pode iniciar uma ou mais tarefas em lote. O agente IBM i monitora estas tarefas em lote, que são referidas como tarefas-filhas.

Ao procurar e monitorar qualquer tarefa-filha que é iniciada, o agente IBM i usa uma alta porcentagem de seu tempo de processamento.

Se você souber que seu planejamento de tarefas não inicia nenhuma tarefa-filha ou se não tiver interesse no monitoramento de tarefas-filhas, será possível instruir o agente IBM i a não procurar e monitorar tarefas-filhas e, portanto, melhorar o desempenho do agente.

É possível excluir o monitoramento de tarefas-filhas no nível do agente para todos os comandos ou no nível da definição de tarefa para um único comando. Se desejar o monitoramento de tarefas-filhas apenas para alguns comandos enviados específicos, é possível configurar esta opção no nível da definição de tarefa para um único comando.

É possível executar um ou dois dos seguintes procedimentos para excluir ou incluir o monitoramento de tarefas-filhas:

Excluir tarefas-filhas do monitoramento de tarefas no nível do agente
Por padrão, as tarefas-filhas são monitoradas. É possível excluir tarefas-filhas do monitoramento para todos os comandos enviados, criando a variável de ambiente do sistema `TWS_NOCHILDS` usando o seguinte comando do sistema IBM i:

```
ADDENVVAR ENVVAR(TWS_NOCHILDS)LEVEL(*SYS)
```

Se o agente IBM i localizar o `TWS_NOCHILDS` no sistema IBM i, ele não monitorará tarefas-filhas para nenhum comando enviado.

Excluir ou incluir tarefas-filhas do monitoramento de tarefas no nível da definição de tarefa
É possível excluir ou incluir tarefas-filhas no monitoramento de tarefas para uma tarefa específica, usando `:NOCHILDS` ou `:CHILDS` como tokens finais da sequência de caracteres de comando para o comando específico.

- Se você incluir o token final `:NOCHILDS` no final do comando nativo que está sendo enviado, o agente IBM i ignorará as tarefas-filhas iniciadas pelo comando.
- Se você incluir o token final `:CHILDS` no final do comando que está sendo enviado, o agente IBM i localizará e monitorará todas as tarefas-filhas iniciadas pelo comando.

Nota: A configuração no nível da definição de tarefa substitui a configuração no nível do agente.

O comando do sistema **SBMJOB**, quando enviado, sempre inicia uma tarefa em lote. Não tente excluir o monitoramento de tarefas, porque se o agente IBM i localizar o comando `SBMJOB` na definição de tarefa, ele removerá e ignorará o token final `:CHILDS` ou `:NOCHILDS` na definição de tarefa e também ignorará a configuração da variável do sistema `TWS_NOCHILDS`.

Exemplos

Para monitorar tarefas-filhas que são iniciadas quando o programa PAYROLL é executado, defina o seguinte comando na definição de tarefa:

- Se a variável do sistema TWS_NOCHILDS estiver definida no sistema IBM i:

  ```
  CALL PGM(MYLIB/PAYROLL):CHILDS
  ```

- Se a variável do sistema TWS_NOCHILDS não estiver definida no sistema IBM i:

  ```
  CALL PGM(MYLIB/PAYROLL)
  ```

Para não monitorar tarefas-filhas que são iniciadas quando o programa MYSCHEDULE é executado, defina o seguinte comando na definição de tarefa:

- Se a variável do sistema TWS_NOCHILDS não estiver definida no sistema IBM i:

  ```
  CALL PGM(MYLIB/MYSCHEDULE):NOCHILDS
  ```

- Se a variável do sistema TWS_NOCHILDS estiver definida no sistema IBM i:

  ```
  CALL PGM(MYLIB/MYSCHEDULE)
  ```

Nota: O comando SBMJOB sempre inicia uma tarefa-filha. O agente IBM i monitora a tarefa-filha mesmo que você defina um comando SBMJOB como na seguinte definição de tarefa:

```
SBMJOB CMD(CALL PGM(MYLIB/USERPGM)):NOCHILDS
```

Informações sobre Monitoramento de Tarefas-filhas nos Logs de Tarefas do Agente IBM i

Se você incluir o monitoramento de tarefas-filhas em agentes IBM i, conforme descrito em "Monitoramento de Tarefas-filhas em Agentes IBM i" na página 644, será possível ver informações relacionadas ao monitoramento de tarefas-filhas no log da tarefa do agente IBM i.

Exemplos

Este exemplo mostra as informações relacionadas ao monitoramento de tarefas-filhas incluído no nível da tarefa para a tarefa CHILDLONG_CHILD no agente NC117025:

```
Job CHILDLONG_CHILD

Workstation (Job) NC117025

Job Stream AS400ENVSET

Workstation (Job Stream) NC117025


===============================================================
= JOB  : NC117025#AS400ENVSET[(1417 10/30/12),(AS400ENVSET)].CHILDLONG_CHILD
= TASK  : <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdlibmi="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdlibmi"
name="ibmi">
  <jsdl:variables>
    <jsdl:stringVariable name="tws.jobstream.id">AS400ENVSET</jsdl:stringVariable>
    <jsdl:stringVariable name="tws.job.workstation">NC117025</jsdl:stringVariable>
    <jsdl:stringVariable name="tws.job.iawstz">201210301417</jsdl:stringVariable>
  </jsdl:variables>
  <jsdl:application name="ibmi">
    <jsdlibmi:ibmi>
      <jsdlibmi:IBMIParameters>
         <jsdlibmi:Task>
          <jsdlibmi:command>CALL PGM(MINERMA/SBM5JOBS) :CHILDS</jsdlibmi:command>
         </jsdlibmi:Task>
      </jsdlibmi:IBMIParameters>
     </jsdlibmi:ibmi>
  </jsdl:application>
  <jsdl:resources>
    <jsdl:orderedCandidatedWorkstations>
      <jsdl:workstation>805E5EAC1F5911E2B9DB6F8202778C47</jsdl:workstation>
    </jsdl:orderedCandidatedWorkstations>
  </jsdl:resources>
</jsdl:jobDefinition>
= TWSRCMAP :
= AGENT  : NC117025
= Job Number: 760232858
= Tue Oct 30 14:16:31 CET 2012
===============================================================
The Dynamic Agent submitter-monitor job is qualified as:
   JobName=DYNAMICMON JobUser=CLAUDIO JobNumber=361743
Here follows the user command string
```

```
   <CALL PGM(MINERMA/SBM5JOBS) :CHILDS>
30/10/2012 14:16:28.844 - Dynamic Agent job submitted the User Command
   CALL PGM(MINERMA/SBM5JOBS)
The FOLLOWING 5 JOBS STARTED under the submitted User Command
  JobName=CLAUDIO JobUser=CLAUDIO JobNumber=361765
  JobName=CLAUDIO JobUser=CLAUDIO JobNumber=361756
  JobName=CLAUDIO JobUser=CLAUDIO JobNumber=361762
  JobName=CLAUDIO JobUser=CLAUDIO JobNumber=361775
  JobName=CLAUDIO JobUser=CLAUDIO JobNumber=361774
Message CPF1241 (Success) received on MsgQueue CLAUDIO QUSRSYS
   for the job CLAUDIO CLAUDIO 361765
Message CPF1241 (Success) received on MsgQueue CLAUDIO QUSRSYS
   for the job CLAUDIO CLAUDIO 361756
Message CPF1241 (Success) received on MsgQueue CLAUDIO QUSRSYS
   for the job CLAUDIO CLAUDIO 361762
Message CPF1241 (Success) received on MsgQueue CLAUDIO QUSRSYS
   for the job CLAUDIO CLAUDIO 361775
Message CPF1241 (Success) received on MsgQueue CLAUDIO QUSRSYS
   for the job CLAUDIO CLAUDIO 361774
*** END codes gathered by the Monitor job ***
 > END Status Code (Status): 0
 > PROGRAM Return Code (Prc): 0
 > USER Return Code (Urc): 0
   Urc was retrieved through SYSAPI

30/10/2012 14:21:37.890 - Dynamic Agent job ended monitoring the User Command
*** Return Code for submitted Command is 0 ***
*** User Command ended successfully ***
```

Este exemplo mostra o log da tarefa para a tarefa `CHILDLING_NC` no agente `NC117025` quando o monitoramento de tarefas-filhas é excluído no nível da tarefa:

```
Job CHILDLING_NC

Workstation (Job) NC117025

Job Stream AS400ENVSET

Workstation (Job Stream) NC117025


============================================================
= JOB  : NC117025#AS400ENVSET[(1417 10/30/12),(AS400ENVSET)].CHILDLING_NC
= TASK  : <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdlibmi="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdlibmi"
name="ibmi">
  <jsdl:variables>
    <jsdl:stringVariable name="tws.jobstream.id">AS400ENVSET</jsdl:stringVariable>
    <jsdl:stringVariable name="tws.job.workstation">NC117025</jsdl:stringVariable>
    <jsdl:stringVariable name="tws.job.iawstz">201210301417</jsdl:stringVariable>
  </jsdl:variables>
  <jsdl:application name="ibmi">
    <jsdlibmi:ibmi>
       <jsdlibmi:IBMIParameters>
         <jsdlibmi:Task>
           <jsdlibmi:command>CALL PGM(MINERMA/SBM5JOBS) :NOCHILDS</jsdlibmi:command>
         </jsdlibmi:Task>
       </jsdlibmi:IBMIParameters>
        </jsdlibmi:ibmi>
  </jsdl:application>
  <jsdl:resources>
    <jsdl:orderedCandidatedWorkstations>
      <jsdl:workstation>805E5EAC1F5911E2B9DB6F8202778C47</jsdl:workstation>
    </jsdl:orderedCandidatedWorkstations>
  </jsdl:resources>
</jsdl:jobDefinition>
= TWSRCMAP :
= AGENT  : NC117025
= Job Number: 760232857
= Tue Oct 30 14:17:01 CET 2012
============================================================
The Dynamic Agent submitter-monitor job is qualified as:
   JobName=DYNAMICMON JobUser=CLAUDIO JobNumber=361817
Here follows the user command string
   <CALL PGM(MINERMA/SBM5JOBS) :NOCHILDS>
30/10/2012 14:16:58.330 - Dynamic Agent job submitted the User Command
   CALL PGM(MINERMA/SBM5JOBS)
As per user choice, NO job started under the submitted command will be monitored
*** END codes gathered by the Monitor job ***
 > END Status Code (Status): 0
 > PROGRAM Return Code (Prc): 0
 > USER Return Code (Urc): 0
   Urc was retrieved through SYSAPI

30/10/2012 14:17:10.220 - Dynamic Agent job ended monitoring the User Command
*** Return Code for submitted Command is 0 ***
*** User Command ended successfully ***
```

A recuperação do código de retorno do agente

O modelo de programação IBM i foi originalmente baseado em um modelo de orientação de objeto anterior no qual os programas se comunicavam através da passagem de mensagens, em vez de usar códigos de retorno. A introdução do modelo Integrated Language Programming (ILE) leva a definições de áreas comuns para dados de troca como códigos de retorno no mesmo ambiente de tarefa: os códigos de retorno do usuário e os códigos finais do sistema.

Para obter informações sobre os códigos de retorno do usuário, consulte "Controlando o Ambiente de Tarefa com o Código de Retorno do Usuário".

Quando o agente IBM i verifica se um comando foi enviado ou uma tarefa foi concluída, ele designa um código de retorno à tarefa com base no status de tarefa da tarefa concluída. O código de retorno é configurado dependendo da mensagem de conclusão do comando ou da tarefa. Se o comando ou tarefa forem concluídos com êxito, o código de retorno será configurado como 0. Se o comando ou tarefa não for concluído com êxito, o código de retorno será configurado como o valor da gravidade da mensagem relacionada à exceção que causou o encerramento anormal da tarefa. O agente IBM i também pode configurar o código de retorno com o valor do código de retorno do usuário quando ele for retornado pelo comando enviado. Se for recuperado, o código de retorno do usuário será usado como o valor para configurar o código de retorno.

O valor de código de retorno designado à tarefa está incluído no log da tarefa do agente IBM i para a tarefa e enviado de volta à interface com o usuário do planejador (painéis WEB UI ou z/OS ISPF) como código de retorno, por razões de compatibilidade com agentes em outros sistemas operacionais.

Controlando o Ambiente de Tarefa com o Código de Retorno do Usuário

Com a introdução do modelo IBM i ILE, é possível recuperar um valor retornado por um programa chamado dentro da mesma tarefa.

Quando o Agent Monitor verifica se um comando enviado foi concluído, ele recupera os seguintes códigos de término da tarefa usando uma interface de programação de aplicativos do Sistema IBM i:

O código de status de término ou <Status> (0 se for bem-sucedido)

Ele indica se o sistema emitiu um cancelamento controlado da tarefa. Os valores possíveis são:

1 o subsistema ou a própria tarefa é cancelada.

0 o subsistema ou a própria tarefa não é cancelada.

blank a tarefa não está em execução.

Código de retorno do programa ou <Prc> (0000 se for bem-sucedido)

Ele especifica o código de conclusão do último programa (como um programa utilitário de arquivo de dados ou um programa RPG ou COBOL, chamado pela tarefa).

Se a tarefa não incluir programa, o código de retorno do programa será 0.

Código de retorno do usuário ou <Urc> (0000 se for bem-sucedido)

Ele especifica o código de retorno definido pelo usuário configurado por construções de linguagem de alto nível ILE. Por exemplo, o código de retorno de um programa escrito em linguagem C.

Ele representa o código de retorno mais recente configurado por algum encadeamento dentro da tarefa.

Se o comando enviado for uma chamada para um programa ILE do usuário retornando um valor na saída, este valor está localizado no código de término da tarefa Urc.

É possível decidir como controlar o ambiente da tarefa de suas tarefas enviadas preparando os comandos para serem enviados como CALLs para seus programas ILE, em que o fluxo interno é controlado e o status de término é decidido por meio de valores de saída adequados. Se um programa do usuário terminar com erro para um controle de fluxo incorreto, sem retornar um valor, o Agent Monitor não configura o Código de Retorno como código de retorno do usuário (Urc), mas segue os critérios descritos em "A recuperação do código de retorno do agente" na página 646.

O exemplo a seguir mostra um programa do usuário C ILE em que duas tarefas em lote são ativadas e um valor igual a 10 é retornado ao responsável pela chamada, independentemente do status de conclusão das tarefas em lote.

```
============================================================================
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
void main(int argc, char *argv[])
{
   int EnvVarRC=0;
   printf("issuing SBMJOB CMD(CALL MYLIB/DIVBY0)...\n");
   system("SBMJOB CMD(CALL MYLIB/DIVBY0)");
   printf("issuing SBMJOB CMD(WRKACTJOB OUTPUT(*PRINT))...\n");
   system("SBMJOB CMD(WRKACTJOB OUTPUT(*PRINT)) LOG(4 0 *SECLVL)");
   exit(10);
   return;
}
============================================================================
```

Método Alternativo para Configurar o Código de Retorno do Usuário

Em alguns ambientes IBM i, a interface de programação de aplicativos do sistema recuperando o código de retorno do usuário (Urc) do código do Agent Monitor não recupera o valor correto para Urc. Assim não é recomendado o uso das APIs do sistema IBM i para recuperar o código de retorno do usuário. Para receber um valor retornado por um programa chamado, é melhor fornecer, em vez disso, um parâmetro para receber o valor.

Mesmo que o Agent Monitor possa recuperar o código de retorno do usuário usando a API do sistema, um método de recuperação de código de retorno de usuário alternativo foi implementado no código do Agent Monitor. O método de recuperação alternativo possui a seguinte lógica. A variável de ambiente da tarefa `USERRC` é criada e configurada como o valor *INI* antes do envio do comando de usuário. Quando o comando é encerrado, o Agent Monitor recupera seu código de retorno de usuário usando as APIs do sistema, mas também verifica se a variável de ambiente da tarefa `USERRC` foi atualizada no nível do programa do usuário. Se um valor diferente de *INI* for localizado, este será considerado como o código de retorno de usuário e o valor recuperado usando as APIs do sistema será ignorado porque o programa do usuário modificou o valor da variável de ambiente da tarefa `USERRC`.

A mudança da variável `USERRC` no nível do programa do usuário requer a mudança do valor `USERRC` antes de sair do código do usuário do aplicativo. No caso do ILE C, é possível fazer isso usando a instrução **putenv**, na qual o código de retorno de usuário é configurado para ser retornado.

O exemplo a seguir mostra como o código do usuário retorna o código de retorno de usuário usando a variável de ambiente da tarefa `USERRC` reservada do agente IBM i. Esse código foi obtido do código do exemplo em "Controlando o Ambiente de Tarefa com o Código de Retorno do Usuário" na página 647 pela substituição de **exit** com a instrução **putenv**.

```
============================================================================
#include <stdio.h>
#include <stdlib.h>
```

```
#include <string.h>
void main(int argc, char *argv[])
{
   int EnvVarRC=0;
   printf("issuing SBMJOB CMD(CALL MYLIB/DIVBY0)...\n");
   system("SBMJOB CMD(CALL MYLIB/DIVBY0)");
   printf("issuing SBMJOB CMD(WRKACTJOB OUTPUT(*PRINT))...\n");
   system("SBMJOB CMD(WRKACTJOB OUTPUT(*PRINT)) LOG(4 0 *SECLVL)");
   EnvVarRC = putenv("USERRC=10");
   return;
}
================================================================================
```

Apêndice A. Evento de Automação e Definições de Ação da Carga de Trabalho Orientada a Evento

Este apêndice documenta os provedores de eventos e ação que podem ser utilizados para a automação da carga de trabalho orientada a evento e fornece detalhes sobre as definições de eventos e ações.

Provedores e Definições de Eventos

Esta seção fornece detalhes sobre os tipos de evento dos seguintes provedores de eventos:

- `TWSObjectsMonitor`
- `FileMonitor`
- `TWSApplicationMonitor`

Data/hora
Contém a data e o horário. É possível especificar um ou ambos os valores no filtro.

Vários predicados de filtro permitidos
É possível especificar vários predicados de filtro para esta propriedade. O evento corresponderá à condição de evento se todos os predicados forem satisfeitos.

Vários valores permitidos
É possível especificar vários valores para esta propriedade em um único predicado de filtro. O filtro será satisfeito quando um dos valores for correspondido.

Caractere curinga permitido
Os curingas suportados são o asterisco (*) e o ponto de interrogação (?).

Eventos do TWSObjectsMonitor

Os eventos do `TWSObjectsMonitor` são:

- Status da Tarefa Alterado
- Tarefa Até
- Tarefa Enviada
- Tarefa Cancelada
- Tarefa Reiniciada
- Tarefa Atrasada
- Tarefa Promovida
- Nível de Risco da Tarefa Alterado
- A Tarefa Excedeu a Duração Máxima
- A Tarefa não Atingiu a Duração Mínima
- Status do Fluxo de Tarefas Alterado
- Fluxo de Tarefas Concluído
- Fluxo de Tarefas Até
- Fluxo de Tarefas Enviado
- Fluxo de Tarefas Cancelado
- Fluxo de Tarefas Atrasado
- Status da Estação de Trabalho Alterado
- Status do Servidor de Aplicativos Alterado
- Link da Estação de Trabalho Filha Alterado
- Link da Estação de Trabalho Pai Alterado

© Copyright IBM Corp. 1999, 2014

- Status do Prompt Alterado
- ProductAlert

Estes eventos são gerados pelo batchman (ou mailman para as estações de trabalho) e gravados em um arquivo de caixa postal denominadomonbox.msg. Os objetos de planejamento são monitorados conforme a seguir:
- As tarefas são monitoradas pela estação de trabalho na qual elas são executadas
- Os fluxos de tarefas são monitorados pelo gerenciador de domínio principal
- As estações de trabalho monitoram a si mesmas
- Os prompts locais são monitorados pela estação de trabalho que está executando a tarefa ou o fluxo de tarefas que possui uma dependência no prompt
- Os prompts globais são monitorados pelo gerenciador de domínio principal

Clique aqui para visualizar os campos Dynamic Workload Console de cada tipo de evento.

Nota: Usuários de PDF, as tabelas de parâmetros acima são um arquivo de HTML referenciado pelo PDF. Ele não está salvo localmente com o PDF do centro de informações. Você deve primeiro visualizá-lo no centro de informações antes de salvá-lo e imprimi-lo.

Trabalhando com eventos WorkstationStatusChanged

O evento é enviado quando uma estação de trabalho é iniciada ou interrompida. Mas existem as seguintes diferenças operacionais dependendo do tipo de estação de trabalho que é monitorada:
- Para uma agente tolerante a falhas, o evento é enviado quando a estação de trabalho é iniciada ou interrompida.
- Para uma estação de trabalhoTivoli dynamic workload broker, o evento também é enviado quando ela é vinculada ou quando o seu link é removido (pois esses comandos também iniciam ou param a estação de trabalho).
- Para uma estação de trabalho de conjunto dinâmico, o evento nunca é enviado (mesmo se o hosting Tivoli dynamic workload broker for interrompido) porque não há monitoramento neste tipo de estação de trabalho.

Exemplos

A regra no exemplo a seguir envia o fluxo de tarefas RJS_102739750 na estação de trabalhoNC125102 assim que todas as tarefas do fluxo de tarefas RCF_307577430 da estação de trabalhoNA022502 estiverem com o status RUNNING ou SUCCESSFUL.

```
<?xml version="1.0"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules"
      xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/
                          event-management/rules/EventRules.xsd">
   <eventRule name="TWS_PLAN_EVENTS_JOB_STATUS_CHANGED"
ruleType="filter" isDraft="no">
      <description>Event: Job Status Changed; Action: Submit
job stream</description>
      <timeZone>Europe/Rome</timeZone>
      <validity from="2011-04-24" to="2012-04-24" />
      <activeTime start="00:00:00" end="12:00:00" />
      <eventCondition name="jobStatChgEvt1"
                      eventProvider="TWSObjectsMonitor"
                      eventType="JobStatusChanged">
      <scope>* # JOBSTREAMVALUE . * [RUNNING,
SUCCESSFUL]</scope>
         <filteringPredicate>
          <attributeFilter name="JobStreamWorkstation" operator="eq">
```

```
         <value>NA022502</value>
          </attributeFilter>
         <attributeFilter name="JobStreamName" operator="eq">
         <value>RCF_307577430</value>
         </attributeFilter>
         <attributeFilter name="JobName" operator="eq">
         <value>*</value>
         </attributeFilter>
         <attributeFilter name="Priority" operator="ge">
         <value>10</value>
         </attributeFilter>
         <attributeFilter name="Monitored" operator="eq">
         <value>true</value>
         </attributeFilter>
         <attributeFilter name="Status" operator="eq">
         <value>Running</value>
         <value>Successful</value>
         </attributeFilter>
         <attributeFilter name="Login" operator="eq">
         <value>TWS_user</value>
         </attributeFilter>
         </filteringPredicate>
      </eventCondition>
      <action actionProvider="TWSAction" actionType="sbs"
responseType="onDetection">
         <description>Launch an existing TWS job
stream</description>
         <scope>SBS NC125102#RJS_102739750</scope>
         <parameter name="JobStreamWorkstationName">
            <value>NC125102</value>
         </parameter>
         <parameter name="JobStreamName">
            <value>RJS_102739750</value>
         </parameter>
      </action>
   </eventRule>
</eventRuleSet>
```

A regra no exemplo a seguir envia a tarefa RJR_30411 na estação de trabalhoNC122160 assim que o fluxo de tarefas RJS_102739750 da estação de trabalhoNC125102 for enviado.

```
<?xml version="1.0"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules"
      xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/
                          event-management/rules/EventRules.xsd">
   <eventRule name="TWS_PLAN_EVENTS_JOB_STREAM_SUBMITTED"
ruleType="filter" isDraft="no">
      <description>Event: Job Stream Submitted; Action: Submit
job</description>
      <eventCondition name="jsSubEvt1"
                      eventProvider="TWSObjectsMonitor"
                      eventType="JobStreamSubmit">
      <scope>WORKSTATIONVALUE # JOBSTREAMVALUE</scope>
         <filteringPredicate>
          <attributeFilter name="JobStreamWorkstation" operator="eq">
         <value>NC125102</value>
          </attributeFilter>
         <attributeFilter name="JobStreamName" operator="eq">
         <value>RJS_102739750</value>
         </attributeFilter>
         <attributeFilter name="Priority" operator="range">
         <value>15</value>
         <value>30</value>
         </attributeFilter>
         <attributeFilter name="LatestStart" operator="le">
         <value>2011-04-26</value>
         </attributeFilter>
         </filteringPredicate>
      </eventCondition>
```

```
      <action actionProvider="TWSAction" actionType="sbj"
responseType="onDetection">
         <description>Launch an existing TWS job
stream</description>
         <scope>SBJ NC122160#RJR_30411 INTO
NC122160#JOBS</scope>
         <parameter name="JobUseUniqueAlias">
            <value>true</value>
         </parameter>
         <parameter name="JobDefinitionName">
            <value>RJR_30411</value>
         </parameter>
         <parameter name="JobDefinitionWorkstationName">
            <value>NC122160</value>
         </parameter>
      </action>
   </eventRule>
</eventRuleSet>
```

Eventos de FileMonitor

Os eventos de FileMonitor são:

- FileCreated
- FileDeleted
- ModificationCompleted
- LogMessageWritten

Ao monitorar arquivos usando os eventos FileCreated, FileDeleted e LogMessageWritten, a memória consumida pelos processos ssmagent.bin e ssmagent.exe aumenta linearmente com o número de arquivos monitorados e com o número de eventos criados. Assim, tenha em mente que o uso mais intenso de caracteres curingas feito dentro desses tipos de eventos, e o número consequentemente mais alto de arquivos monitorados, resultará em um consumo de memória mais intenso pelos processos ssmagent.bin e ssmagent.exe.

Eventos FileMonitor não são suportados em:

- Conjuntos, conjuntos dinâmicos e estações de trabalho de mecanismo remoto.
- IBM i.

Clique aqui para visualizar os campos Dynamic Workload Console para cada tipo de evento.

Nota: Usuários de PDF, as tabelas de parâmetros acima são um arquivo de HTML referenciado pelo PDF. Ele não está salvo localmente com o PDF do centro de informações. Você deve primeiro visualizá-lo no centro de informações antes de salvá-lo e imprimi-lo.

Usando a propriedade MatchExpression da regra de evento LogMessageWritten

O plug-in de evento LogMessageWritten usa a expressão regular especificada na propriedade MatchExpression para executar correspondências de subsequência em entradas nos arquivos de log que estão sendo monitorados. O valor de MatchExpression deve ser uma expressão regular válida de acordo com as regras de sintaxe de expressão regular do agente Netcool/SSM que o plug-in de eventos usa.

A tabela a seguir descreve a sintaxe dos tokens de expressão regular suportada pelo Netcool/SSM. Observe que para escrever uma expressão regular válida para a propriedade MatchExpression você deve escrever o caractere de escape \ (barra

invertida) antes de cada token usado na sintaxe da expressão regular (por exemplo, \^ ou \$). Quando o token já especificar um caractere de barra invertida, você deve escrever os caracteres de barra invertida (por exemplo, \\< ou \\b).

Tabela 109. Sintaxe da expressão regular.

Token	Corresponde
.	Qualquer caractere.
^	O início de uma linha (uma sequência de comprimento zero).
$	O final de uma linha. Uma nova linha ou o final do buffer de procura.
\<	O início de uma palavra (no qual uma palavra é uma sequência de caracteres alfanuméricos).
\>	O final de uma palavra (a sequência de comprimento zero entre um caractere alfanumérico e um caractere não alfanumérico).
\b	Qualquer limite de palavra (isto é equivalente a (\<¦\>)).
\d	Um caractere de dígito.
\D	Qualquer caractere que não seja um dígito.
\w	Um caractere de palavra (alfanumérico ou sublinhado).
\W	Qualquer caractere que não seja um caractere de palavra (alfanumérico ou sublinhado).
\s	Um caractere de espaço em branco.
\S	Qualquer caractere que não seja um espaço em branco.
\c	Caracteres especiais ou de escape. Os caracteres a seguir são interpretados de acordo com as convenções da linguagem C: \0, \a, \f, \n, \r, \t, \v. Para especificar um caractere em hexadecimal, use a sintaxe \xNN. Por exemplo, \x41 é o caracter ASCII A.
\	Todos os caracteres separados desses descritos acima podem ser codificados com a função escape usando o prefixo de barra invertida. Por exemplo, para especificar o uso de um colchete esquerdo simples \[.
[]	Qualquer um dos caracteres especificados em um conjunto. Um conjunto explícito de caracteres pode ser especificado como em [aeiou] assim como intervalos de caracteres como [0-9A-Fa-f], que correspondem a qualquer dígito hexadecimal. O traço (-) perde seu significado especial quando codificado com a função escape, como em [A\-Z] ou quando ele for o primeiro ou o último caractere em um conjunto, como em [-xyz0-9]. Todas as regras de escape acima com a barra invertida podem ser usadas entre []. Por exemplo, a expressão [\x41-\x45] é equivalente a [A-D] em ASCII. Para usar um colchete de fechamento em um conjunto, codifique-o com a função escape usando [\]] ou utilize-o como o primeiro caractere no conjunto, como []xyz]. Classes de caractere estilo POSIX também são permitidas em um conjunto de caracteres. A sintaxe para classes de caractere é [:class:]. As classes de caractere suportadas são: • [:alnum:] - caracteres alfanuméricos. • [:alpha:] - caracteres alfabéticos. • [:blank:] - caracteres de espaço e de tabulação. • [:cntrl:] - caracteres de controle. • [:digit:] - caracteres numéricos. • [:graph:] - caracteres que são tanto para impressão quanto visualização. • [:lower:] - caracteres alfabéticos em minúsculas. • [:print:] - caracteres para impressão (caracteres que não são caracteres de controle). • [:punct:] - caracteres de pontuação (caracteres que não são letras, dígitos, caracteres de controle ou espaços). • [:space:] - caracteres de espaço (como caracteres de espaço, de tabulação e de feed de formulário). • [:upper:] - caracteres alfabéticos maiúsculos. • [:xdigit:] - caracteres que são dígitos hexadecimais. Os colchetes são permitidos dentro dos colchetes do conjunto. Por exemplo, [a-z0-9!] é equivalente a [[:lower:][:digit:]!] no código de idioma C.
[^]	Inverte o comportamento de um conjunto de caracteres [] como descrito acima. Por exemplo, [^[:alpha:]] corresponde a qualquer caractere que não seja alfabético. O símbolo de circunflexo ^ apenas possui este significado especial quando for o primeiro caractere em um conjunto de colchetes.
{n}	Exatamente n ocorrências da expressão anterior, em que 0 <= n <= 255. Por exemplo, a{3} corresponde a aaa.
{n,m}	Entre n e m ocorrências da expressão anterior, em que 0 <= n <= m <= 255. Por exemplo, um número hexadecimal de 32 bits pode ser descrito como 0x[[:xdigit:]]{1,8}.
{n,}	No mínimo n ou mais (até o infinito) ocorrências da expressão anterior.
*	Nenhuma expressão anterior ou mais de uma expressão anterior.
+	Uma expressão anterior ou mais expressões anteriores.
?	Nenhuma ou uma expressão anterior.
(exp)	Agrupamento; qualquer série de expressões pode ser agrupada entre parênteses para aplicação de um postfix ou um operador de barra (¦) para um grupo de expressões sucessivas. Por exemplo: • ab+ corresponde a todos os abbb • (ab)+ corresponde a todos os ababab

Tabela 109. Sintaxe da expressão regular. (continuação)

Token	Corresponde
¦	Expressões alternativas (OR lógico). A barra vertical (¦) tem precedência inferior a todos os tokens na linguagem da expressão regular. Isso significa que ab¦cd corresponde a todos os cd mas não corresponde a abd (neste caso use a(b¦c)d).

Dica: Ao definir expressões regulares para corresponderem a caracteres com diversos bytes, envolva cada caractere com diversos bytes entre parênteses ().

O Tabela 110 fornece um conjunto de exemplos de expressões regulares, junto com as sequências de amostra assim como os resultados de aplicação da expressão regular para essas sequências.

Há dois casos importantes em expressões regulares com sequências. Uma expressão regular pode corresponder a uma sequência inteira (um caso conhecido como uma *correspondência de sequência*) ou apenas a uma parte dessa sequência (um caso conhecido como uma *correspondência de subsequência*). Por exemplo, a expressão regular `\<int\>` irá gerar uma correspondência de sequência para a sequência `int x` mas não irá gera uma correspondência de sequência. Essa distinção é importante porque alguns subagentes não suportam correspondência de subsequência. Onde for aplicável, os resultados listados nos exemplos diferenciam entre correspondências de sequência e subsequência.

Tabela 110. Exemplos de expressão regular.

Esta expressão...	Aplicada a esta sequência...	Resulta em...
.	a	Correspondência de sequência
	!	Correspondência de sequência
	abcdef	Correspondência de subsequência em a
	sequência vazia	Sem correspondência
M..COUNT	MINCOUNT	Correspondência de sequência
	MXXCOUNTY	Correspondência de subsequência em MXXCOUNT
	NONCOUNT	Sem correspondência
.*	sequência vazia	Correspondência de sequência
	Animal	Correspondência de sequência
.+	Qualquer sequência não vazia	Correspondência de sequência
	sequência vazia	Sem correspondência
^	sequência vazia	Correspondência de sequência
	hello	A correspondência de subsequência de comprimento 0 na posição 0 (posição 0 = primeiro caractere na sequência)
$	sequência vazia	Correspondência de sequência
	hello	Correspondência de subsequência de comprimento 0 na posição 5 (a posição 0 = primeiro caractere na sequência)
^$	sequência vazia	Correspondência de sequência
	hello	Sem correspondência
\bee	tee	Sem correspondência
	Paid fee	Sem correspondência
	feel	Sem correspondência
	eel	Correspondência de subsequência em ee
.*thing.*	The thing is in here	Correspondência de sequência
	there is a thing	Correspondência de sequência
	it isn't here	Sem correspondência
	thinxxx	Sem correspondência
a*	sequência vazia	Correspondência de sequência
	aaaaaaaaa	Correspondência de sequência
	a	Correspondência de sequência
	aardvark	Correspondência de subsequência em aa
	esta sequência	Correspondência de subsequência
((ab)*c)*	sequência vazia	Correspondência de sequência
	cccccccc	Correspondência de sequência
	ccccabcccabc	Correspondência de sequência

Tabela 110. Exemplos de expressão regular. (continuação)

Esta expressão...	Aplicada a esta sequência...	Resulta em...
a+	sequência vazia	Sem correspondência
	aaaaaaaaa	Correspondência de sequência
	a	Correspondência de sequência
	aardvark	Correspondência de subsequência em aa
	esta sequência	Sem correspondência
(ab)+c)*	sequência vazia	Correspondência de sequência
	abababababcabc	Correspondência de sequência
(ab){2}	abab	Correspondência de sequência
	cdabababab	Correspondência de subsequência em abab
[0-9]{4,}	123	Sem correspondência
	a1234	Correspondência de subsequência em 1234
a{0}	sequência vazia	Correspondência de sequência
	a	Sem correspondência
	hello	A correspondência de subsequência de comprimento 0 na posição 0 (posição 0 = primeiro caractere na sequência)
[0-9]{1,8}	isto não é um número	Sem correspondência
	a=4238, b=4392876	Correspondência de subsequência em 4238
([aeiou][^aeiou])+	Hello	Correspondência de subsequência em el
	!!! Supacalafraglistic	Correspondência de subsequência em upacalaf
[+-]?1	1	Correspondência de sequência
	+1	Correspondência de sequência
	-1	Correspondência de sequência
	.1	Correspondência de subsequência em 1
	value+1	Correspondência de subsequência em +1
a¦b	a	Correspondência de sequência
	b	Correspondência de sequência
	c	Sem correspondência
	Daniel	Correspondência de subsequência em a
abcd¦efgh	abcd	Correspondência de sequência
	efgh	Correspondência de sequência
	abcdfgh	Correspondência de subsequência em abcd
[0-9A-F]+	BAADF00D	Correspondência de sequência
	F	Correspondência de sequência
	baadF00D	Correspondência de subsequência em F00D
	c	Sem correspondência
	O	Sem correspondência
	g	Sem correspondência
x = \d+	x = 1234	Correspondência de sequência
	x = 0	Correspondência de sequência
	x = 1234a	Correspondência de subsequência em x = 1234
	x = y	Sem correspondência
	x^=^ onde ^ representa um caractere de espaço	Sem correspondência
\D\d	a1	Correspondência de sequência
	a11	Correspondência de subsequência em a1
	-9	Correspondência de sequência
	a	Sem correspondência
	8	Sem correspondência
	aa	Sem correspondência
	4t	Sem correspondência
\s+	Hello_w0rld	Sem correspondência
	Hello^^^world em que ^ representa um caractere de espaço	Correspondência de subsequência em ^^^ em que ^ representa um caractere de espaço
	Widget^ em que ^ representa um caractere de espaço	Correspondência de subsequência em ^ em que ^ representa um caractere de espaço
	^^^^^ em que ^ representa um caractere de espaço	Correspondência de sequência

Tabela 110. Exemplos de expressão regular. (continuação)

Esta expressão...	Aplicada a esta sequência...	Resulta em...
\S+	Hello_w0rld	Correspondência de subsequência de comprimento 11 em Hello_w0rld
	Hello^^^world em que ^ representa um caractere de espaço	Correspondência de subsequência em Hello
	Widget^ em que ^ representa um caractere de espaço	Correspondência de subsequência em Widget
	^^^^^ em que ^ representa um caractere de espaço	Sem correspondência
\w+	D4n_v4n Vugt	Correspondência de subsequência em D4n_v4n
	^^^hello em que ^ representa um caractere de espaço	Correspondência de subsequência em hello
	blah	Correspondência de sequência
	x#1	Sem correspondência
	foo bar	Sem correspondência
\W	Hello there	Correspondência de subsequência de comprimento 1 na separação do caractere de espaço
	~	Correspondência de sequência
	aa	Sem correspondência
	a	Sem correspondência
	-	Sem correspondência
	^^^^444 == 5 em que ^ representa um caractere de espaço	Correspondência de subsequência de comprimento 1 no primeiro^ em que ^ representa um caractere de espaço
\w+\s*=\s*\d+	x = 123	Correspondência de sequência
	count0=555	Correspondência de sequência
	my_var=66	Correspondência de sequência
	0101010=0	Correspondência de sequência
	xyz = e	Sem correspondência
	delta=	Sem correspondência
	==8	Sem correspondência
[[:alnum:]]+	1234	Correspondência de sequência
	...D4N13L	Correspondência de subsequência em D4N13L
[[:alpha:]]+	Bubble	Correspondência de sequência
	...DANI3L	Correspondência de subsequência em DANI
	69	Sem correspondência
[:blank:]]+	alpha^^^^and beta em que ^ representa um caractere de espaço	Correspondência de subsequência em ^^^^ em que ^ representa um caractere de espaço
	Animal	Sem correspondência
	sequência vazia	Sem correspondência
[[:space:]]+	alpha^^^^and beta em que ^ representa um caractere de espaço	Correspondência de subsequência em ^^^^ em que ^ representa um caractere de espaço
	Animal	Sem correspondência
	sequência vazia	Sem correspondência
[[:cntrl:]]+	...Hello W0rld!	Sem correspondência
	sequência vazia	Sem correspondência
[[:graph:]]+	hello world	Correspondência de subsequência em hello
	^^^^^ em que ^ representa um caractere de espaço	Sem correspondência
	^^^!? em que ^ representa um caractere de espaço	Correspondência de subsequência em !?
[:lower:]]+	Animal	Correspondência de subsequência em nimal
	ABC	Sem correspondência
	0123	Sem correspondência
	foobar	Correspondência de sequência
	^^^0blaH! em que ^ representa um caractere de espaço	Correspondência de subsequência em bla
[_[:lower:]]+	foo_bar	Correspondência de sequência
	this_thinG!!!	Correspondência de subsequência em _thin
[[:upper:]]+	SIM	Correspondência de sequência
	#define MAX 100	Correspondência de subsequência em MAX
	f00 b4r	Sem correspondência
[[:print:]]+	hello world	Correspondência de sequência
	^^^^^ em que ^ representa um caractere de espaço	Correspondência de sequência

Tabela 110. Exemplos de expressão regular. (continuação)

Esta expressão...	Aplicada a esta sequência...	Resulta em...
[[:punct:]]+	didn't	Correspondência de subsequência em '
	Animal	Sem correspondência
[[:xdigit:]]+	43298742432392187ffe	Correspondência de sequência
	x = bAAdF00d	Correspondência de subsequência em bAAdF00d
	4327afeffegokpoj	Correspondência de subsequência em 4327afeffe
c:\\temp	c:\temp	Correspondência de sequência

Exemplo

A regra no exemplo a seguir envia um e-mail a uma lista de destinatários assim que o arquivo /home/book.txt for criado na estação de trabalho editor_wrkstn.

```
<?xml version="1.0"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules"
      xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/
                          event-management/rules/EventRules.xsd">
   <eventRule name="FILE_MONITOR_FILE_CREATED" ruleType="filter"
isDraft="no">
      <description>Event: File Created; Action: Send
mail</description>
      <validity to="2012-04-22" />
      <eventCondition name="fileCrtEvt1"
eventProvider="FileMonitor" eventType="FileCreated">
      <scope>/HOME/BOOK.TXT ON EDITOR_WRKSTN</scope>
         <filteringPredicate>
          <attributeFilter name="FileName" operator="eq">
         <value>/home/book.txt</value>
          </attributeFilter>
         <attributeFilter name="SampleInterval" operator="eq">
         <value>60</value>
         </attributeFilter>
         <attributeFilter name="Workstation" operator="eq">
         <value>editor_wrkstn</value>
         </attributeFilter>
         <attributeFilter name="Hostname" operator="eq">
         <value>ceditor</value>
         </attributeFilter>
                  </filteringPredicate>
      </eventCondition>
      <action actionProvider="MailSender" actionType="SendMail" responseType="onDetection">
         <description>Send an eMail</description>
         <scope>SAUL.FELLOW@US.IBM.COM, ISAAC.LINGER@US.IBM.COM
: THE EXPECTED FILE
                 HAS BEEN CREATED!</scope>
         <parameter name="Cc">
            <value>william.waulkner@us.ibm.com</value>
         </parameter>
         <parameter name="Bcc">
            <value>ernest.demingway@us.ibm.com</value>
         </parameter>
         <parameter name="Body">
            <value>The expected file was created!
                   The book is ready to be published.</value>
         </parameter>
         <parameter name="To">
            <value>saul.fellow@us.ibm.com,
isaac.linger@us.ibm.com</value>
         </parameter>
         <parameter name="Subject">
            <value>The expected file was created!</value>
         </parameter>
      </action>
   </eventRule>
</eventRuleSet>
```

Eventos do TWSApplicationMonitor

Eventos do TWSApplicationMonitor referem-se a processos, sistema de arquivos e caixa de mensagens do Tivoli Workload Scheduler. São eles:

- MessageQueuesFilling

- TivoliWorkloadSchedulerFileSystemFilling
- TivoliWorkloadSchedulerProcessNotRunning

Eventos TWSApplicationMonitor não são suportados em sistemas IBM i.

Clique aqui para ver os campos Dynamic Workload Console para cada tipo de evento.

Nota: Usuários de PDF, as tabelas de parâmetros acima são um arquivo de HTML referenciado pelo PDF. Ele não está salvo localmente com o PDF do centro de informações. Você deve primeiro visualizá-lo no centro de informações antes de salvá-lo e imprimi-lo.

Exemplo

A regra no exemplo a seguir faz o log da mensagem de aviso LOGMSG01W assim que os arquivos de fila de mensagens intercom ou mailbox na estação de trabalho NC122160 atingirem 70 por cento de seu tamanho.

```
<?xml version="1.0"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules"
      xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/
                          event-management/rules/EventRules.xsd">
   <eventRule name="TWS_APPL_MONITOR_MESSAGE_QUEUES_FILLING"
ruleType="filter" isDraft="no">
      <description>Event: Message queues filling; Action:
Message logger</description>
      <timeZone>America/Los_Angeles</timeZone>
      <validity from="2011-04-25"/>
      <activeTime end="17:00:00"/>
      <eventCondition name="twsMesQueEvt1"
eventProvider="TWSApplicationMonitor"
 eventType="TWSMessageQueues">
      <scope>INTERCOM, MAILBOX FILLED UP 70% ON
NC122160</scope>
         <filteringPredicate>
          <attributeFilter name="MailboxName" operator="eq">
         <value>intercom</value>
         <value>mailbox</value>
          </attributeFilter>
         <attributeFilter name="FillingPercentage"
operator="ge">
         <value>70</value>
         </attributeFilter>
         <attributeFilter name="Workstation" operator="eq">
         <value>NC122160</value>
         </attributeFilter>
         <attributeFilter name="SampleInterval" operator="eq">
         <value>60</value>
         </attributeFilter>
                  </filteringPredicate>
      </eventCondition>
      <action actionProvider="MessageLogger" actionType="MSGLOG"
responseType="onDetection">
         <description>Escreve um log de mensagens de
aviso</description>
         <scope>OBJECT=LOGMSG01W MESSAGE=MAILBOX AND/OR
INTERCOM QUEUE
                HAS REACHED 70% OF FILLING</scope>
         <parameter name="ObjectKey">
            <value>LOGMSG01W</value>
         </parameter>
         <parameter name="Message">
            <value>Caixa de correio e/ou fila de
Intercomunicação atingiu 70% de preenchimento</value>
         <parameter name="Severity">
            <value>Aviso</value>
      </action>
   </eventRule>
</eventRuleSet>
```

Provedores e Definições de Ação

Esta seção fornece detalhes sobre os tipos de ação dos seguintes provedores de ação:

- GenericAction
- MailSender
- MessageLogger
- SmartCloud Control Desk
- TBSMEventForwarder
- TECEventForwarder
- TWSAction
- "TWSForZosAction" na página 664

Ações de GenericAction

Este provedor implementa uma única ação chamada `RunCommand` que executa comandos não Tivoli Workload Scheduler. Os comandos são executados no mesmo computador em que o processador de eventos é executado.

Somente *TWS_user* está autorizado a executar o comando.

Importante: Quando o comando inclui redirecionamento de saída (embora o uso de um ou dois sinais de >), insira o comando em um arquivo executável e configure o nome do arquivo como o argumento da propriedade `Command`.

Clique aqui para visualizar os campos Dynamic Workload Console para `RunCommand`.

Nota: Usuários de PDF, as tabelas de parâmetros acima são um arquivo de HTML referenciado pelo PDF. Ele não está salvo localmente com o PDF do centro de informações. Você deve primeiro visualizá-lo no centro de informações antes de salvá-lo e imprimi-lo.

Exemplo

A regra no seguinte exemplo executa o comando **ps -ef** para listar todos os processos em execução no momento em uma estação de trabalho UNIX quando um parâmetro inválido for encontrado nessa estação de trabalho. Observe que a regra é baseada em um evento customizado desenvolvido usando o provedor de eventos `GenericEventPlugIn`. Para obter informações adicionais sobre tipos de eventos customizados de desenvolvimento, consulte "Definindo Eventos Customizados" na página 134.

```
<?xml version="1.0"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules"
      xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/
                          event-management/rules/EventRules.xsd">
   <eventRule name="CUSTOM_EVENT_GENERIC_EVENT" ruleType="filter"
isDraft="yes">
      <description>Event: Generic Event; Action: Run
Command</description>
      <activeTime start="08:30:00" end="17:30:00"/>
      <eventCondition name="genericEvt3"
eventProvider="GenericEventPlugIn"
      eventType="Event1">
      <scope>INVALID PARAMETER ON
WORKSTATIONVALUE</scope>
         <filteringPredicate>
           <attributeFilter name="Param1" operator="ne">
            <value>Invalid Parameter</value>
```

```
            </attributeFilter>
            <attributeFilter name="Workstation" operator="eq">
             <value>WorkstationValue</value>
            </attributeFilter>
          </filteringPredicate>
        </eventCondition>
        <action actionProvider="GenericActionPlugin"
actionType="RunCommand"
        responseType="onDetection">
           <description>Run a command</description>
           <scope>PS -EF</scope>
           <parameter name="Command">
              <value>ps -ef</value>
           </parameter>
           <parameter name="WorkingDir">
              <value>/home</value>
           </parameter>
        </action>
     </eventRule>
</eventRuleSet>
```

Ações de MailSender

Este provedor implementa uma única ação chamada `SendMail` que se conecta a um servidor SMTP para enviar um e-mail. Use o `optman` para customizar os seguintes atributos relacionados (para obter informações detalhadas sobre o `optman`, consulte o *Guia de Administração*):

- Emissor de correio
- Servidor SMTP
- Número da porta SMTP
- Nome do usuário de correio
- Senha do usuário de correio
- SSL

Clique aqui para visualizar os campos Dynamic Workload Console para `SendMail`.

Nota: Usuários de PDF, a tabela de parâmetros acima é um arquivo de HTML referenciado pelo PDF. Ele não está salvo localmente com o PDF do centro de informações. Você deve primeiro visualizá-lo no centro de informações antes de salvá-lo e imprimi-lo.

Ações de MessageLogger

Este provedor implementa uma única ação denominada `MSGLOG` que registra a ocorrência de uma situação em um banco de dados de auditoria interno. O número de entrada no banco de dados de auditoria é configurável. Existe uma limpeza automática baseada em uma política FIFO.

Clique aqui para visualizar os campos Dynamic Workload Console para `MSGLOG`.

Nota: Usuários de PDF, a tabela de parâmetros acima é um arquivo de HTML referenciado pelo PDF. Ele não está salvo localmente com o PDF do centro de informações. Você deve primeiro visualizá-lo no centro de informações antes de salvá-lo e imprimi-lo.

Ações SmartCloud Control Desk

Este provedor implementa uma única ação, denominada `OpenTicket`, que abre um chamado em um SmartCloud Control Desk padrão. Utilize `optman` para especificar o servidor SmartCloud Control Desk, configurando as opções globais `sccdUrl`, `sccdName` e `sccdUserPassword`. Para obter informações detalhadas sobre o `optman`, consulte o *Guia de Administração*.

Clique aqui para ver os campos Dynamic Workload Console para `OpenTicket`.

Nota: Usuários de PDF, a tabela de parâmetros acima é um arquivo de HTML referenciado pelo PDF. Ele não está salvo localmente com o PDF do centro de informações. Você deve primeiro visualizá-lo no centro de informações antes de salvá-lo e imprimi-lo.

Ações TBSMEventForwarder

Este provedor implementa uma única ação denominada `TBSMFWD` que encaminha o evento para um servidor Tivoli Business Systems Manager externo (ou qualquer outro aplicativo capaz de atender eventos no formato do TBSM, por exemplo, Netcool/OMNIbus). O provedor usa um servidorAnálise EIF padrão, cujo nome do host e porta são definidos pela configuração das opções globais `TECServerName` e `TECServerPort` com `optman`.

Nota: `TECServerName` e `TECServerPort` são ambos usados para aplicativos que processam eventos no formato TEC ou TBSM. Para obter informações detalhadas sobre o `optman`, consulte o *Guia de Administração*.

A Análise IEF usada como destinatário pode ser substituída pelas configurações da ação.

Clique aqui para visualizar os campos Dynamic Workload Console para `TBSMFWD`.

Nota: Usuários de PDF, a tabela de parâmetros acima é um arquivo de HTML referenciado pelo PDF. Ele não está salvo localmente com o PDF do centro de informações. Você deve primeiro visualizá-lo no centro de informações antes de salvá-lo e imprimi-lo.

Configurando o Tivoli Business Services Manager para receber eventos

Para configurar o Tivoli Business Systems Manager para receber eventos do Tivoli Workload Scheduler no Análise EIF, você deve copiar o arquivo `utilities/tivoli_eif_tws.rules` que é fornecido com o DVD do Tivoli Workload Scheduler. Em seguida, você deve editar o arquivo `tivoli_eif.rules` incluindo a seguinte linha:

```
include "tivoli_eif_tws.rules"
```

Ações do TECEventForwarder

Este provedor implementa uma única ação, denominada`TECFWD`, que encaminha o evento para um servidor Tivoli Enterprise Console externo (ou qualquer outro aplicativo capaz de receber eventos no formato TEC). O provedor usa um servidorAnálise EIF padrão, cujo nome do host e porta são definidos pela configuração das opções globais `TECServerName` e `TECServerPort` com `optman`. Para obter informações detalhadas sobre o `optman`, consulte o *Guia de Administração*.

O TEC utilizado como destinatário pode ser substituído pelas configurações de ação.

Clique aqui para ver os campos Dynamic Workload Console para `TECFWD`.

Nota: Usuários de PDF, a tabela de parâmetros acima é um arquivo de HTML referenciado pelo PDF. Ele não está salvo localmente com o PDF do centro de informações. Você deve primeiro visualizá-lo no centro de informações antes de salvá-lo e imprimi-lo.

Ações de TWSAction

As ações de `TWSAction` são:
- SubmitJobStream
- SubmitJob
- SubmitAdHocJob
- ReplyPrompt

Clique aqui para ver os campos Dynamic Workload Console de cada tipo de ação.

Nota: Usuários de PDF, as tabelas de parâmetros acima são um arquivo de HTML referenciado pelo PDF. Ele não está salvo localmente com o PDF do centro de informações. Você deve primeiro visualizá-lo no centro de informações antes de salvá-lo e imprimi-lo.

Usando a propriedade `SchedTimeResolutionCriteria` da ação `SubmitJob`

Esta propriedade é usada para corresponder a tarefa na questão com uma instância específica do fluxo de tarefas que a contém (definida com a propriedade `JobStreamName`) com base no tempo planejado no fluxo de tarefas. Os valores possíveis que podem ser configurados são:

Anterior
A tarefa é enviada com a instância do fluxo de tarefas anterior mais próximo no plano.

Próxima
A tarefa é enviada com a instância do fluxo de tarefas seguinte mais próximo no plano.

Qualquer
A tarefa é enviada com qualquer das instâncias do fluxo de tarefas anterior mais próximo ou seguinte mais próximo no plano.

TWSForZosAction

Este provedor implementa uma única ação chamada `AddJobStream` que inclui uma ocorrência do aplicativo (fluxo de tarefas) no plano atual no IBM Tivoli Workload Scheduler for z/OS. Esse provedor é para uso nas configurações de planejamento de ponta a ponta do Tivoli Workload Scheduler.

A descrição de aplicativo da ocorrência a ser incluída deve existir no banco de dados AD database do IBM Tivoli Workload Scheduler for z/OS.

Clique aqui para visualizar os campos Dynamic Workload Console para `AddJobStream`.

Nota: Usuários de PDF, a tabela de parâmetros acima é um arquivo de HTML referenciado pelo PDF. Ele não está salvo localmente com o PDF do centro de informações. Você deve primeiro visualizá-lo no centro de informações antes de salvá-lo e imprimi-lo.

Exemplo

Nesse exemplo, uma empresa farmacêutica usa a regra ZOSRULE031 para produzir um planejamento de distribuição do merchandise sob o controle do departamentoDISTR07. Assim que a lista de merchandise ordenada que será entregue no mês seguinte estiver pronta e colocada no arquivo MONTHLYORDERS.TXT no agente RU192298 no escritório de uma filial, o sistema centralizado incluirá o aplicativo (fluxo de tarefas) ADFIRST no plano atual. ADFIRST contém as operações (tarefas) que produzem um planejamento de entrega otimizada para o próximo mês.

```
<?xml version="1.0"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules"
      xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/
                          event-management/rules/EventRules.xsd">
   <eventRule name="ZOSRULE031" ruleType="filter" isDraft="no">
      <eventCondition name="fileCrtEvt19" eventProvider="FileMonitor"
      eventType="FileCreated">
      <scope>/PRODORDER/MONTHLYORDERS.TXT ON RU192298</scope>
         <filteringPredicate>
           <attributeFilter name="Param1" operator="ne">
            <value>/prodorder/monthlyorders.txt</value>
           </attributeFilter>
           <attributeFilter name="SampleInterval" operator="eq">
            <value>60</value>
           </attributeFilter>
           <attributeFilter name="Workstation" operator="eq">
           <value>RU192298</value>
           </attributeFilter>
         </filteringPredicate>
      </eventCondition>
      <action actionProvider="TWSForZosAction" actionType="AddJobStream"
      responseType="onDetection">
        <scope>
          ADD JOBSTREAM ADFIRST[DEADLINE OFFSET: 0001] WITH OWNER DISTR07 IN PLAN
        </scope>
         <parameter name="HoldAll">
            <value>false</value>
         </parameter>
         <parameter name="Priority">
            <value>5</value>
         </parameter>
         <parameter name="JobStreamDeadlineOffset">
            <value>0001</value>
         </parameter>
         <parameter name="JobStreamName">
            <value>ADFIRST</value>
         </parameter>
         <parameter name="OwnerDescription">
            <value>Owner description</value>
         </parameter>
         <parameter name="Owner">
            <value>distr07</value>
         </parameter>
         <parameter name="DependenciesResolution">
            <value>All</value>
         </parameter>
         <parameter name="AuthorityGroup">
            <value>AuthGrpBase</value>
         </parameter>
         <parameter name="Parm_1">
            <value>var1=value1</value>
         </parameter>
         <parameter name="Parm_2">
            <value>var2=value2</value>
         </parameter>
         <parameter name="JCLVariableTable">
            <value>VarTableZos01</value>
```

```
            </parameter>
            <parameter name="JobStreamDescription">
               <value>Este fluxo de tarefas contém tarefas que processam ordens para o
                      proprietário DISTR07.</value>
            </parameter>
            <parameter name="Group">
               <value>GroupBase</value>
            </parameter>
         </action>
      </eventRule>
   </eventRuleSet>
```

Apêndice B. Referência do Esquema Job Submission Description Language

Essa seção de referência especifica a semântica e estrutura de Job Submission Description Language (JSDL) que se aplicam especificamente para uso com o Dynamic Workload Broker. O esquema JSDL é usado para descrever vos requisitos de tarefa para envio para recursos. O Dynamic Workload Broker analisa o ambiente de TI e designa o melhor recurso disponível para executar a tarefa, com base nos requisitos especificados.

Introdução

Job Submission Description Language (JSDL) é uma linguagem para descrever os requisitos de tarefa para envio para recursos. A linguagem JSDL contém um vocabulário e esquema XML normativo que facilita a expressão desses requisitos como um conjunto de elementos XML.

Arquivos JSDL seguem a sintaxe e semântica XML, conforme definido no esquema JSDL.

Estrutura do Documento Job Submission Description Language

Um arquivo JSDL é descrito usando a sintaxe XML e segue a sintaxe e semântica XML. A sintaxe XML é um padrão de mercado e não e explicado neste manual. O arquivo JSDL também segue regras de sintaxe JSL específicas, conforme explicado em "Tipos de Elementos Job Submission Description Language" na página 669 e em "Elementos JSDL" na página 672.

O arquivo JSDL consiste em elementos (complexos ou simples) e tipos. Elementos complexos contêm outros elementos enquanto que elementos simples não contêm nenhum outro elemento. Uma especificação de tipo executa uma verificação de sintaxe no valor especificado para o elemento ao qual se refere. Por exemplo, o elemento **physicalMemory** segue o tipo jsdl:NumericRangeType. O tipo jsdl:NumericRangeType especifica que é possível designar a esse elemento um valor numérico específico ou um valor de intervalo numérico. Nenhum outro tipo de valor é suportado para o elemento **physicalMemory**.

O arquivo JSDL é organizado em uma estrutura hierárquica, em que o elemento **jobDefinition** é o elemento-raiz. O elemento **jobDefinition** contém todos os elementos que descrevem a tarefa e seus atributos.

A definição de pseudo esquema tem a seguinte aparência:

```
< jobDefinition >
  <annotation ... />?
  <category>... />*
  <variables ... />?
  <application ... />
  <resources ... />?
  <relatedResources ... />*
  <optimization ... >?
  <scheduling ...>?
</jobDefinition>
```

© Copyright IBM Corp. 1999, 2014

A Tabela 111 fornece uma visualização de tabela do arquivo JSDL indicando os relacionamentos hierárquicos entre os elementos contidos no elemento **jobDefinition**.

Tabela 111. Estrutura Hierárquica do Aqruivo JSDL

Primeiro nível	Segundo nível	Terceiro nível	Quarto nível
annotation			
category			
variables	stringVariable		
	uintVariable		
	doubleVariable		
aplicativo	script		
	argumentos	valor	
	environment	variable name	
	credencial	username	
		groupname	
		password	
	j2ee	invoker	tipo
		jms	connFactory
			destino
			message
		ejb	jndiHome
		credencial	userName
			password
			JAASalias
recursos físicos	candidateHosts	hostName	
	candidateCPUs	cpu	speed
	physicalMemory		
	virtualMemory		
	candidateOperating Systems	operatingsystems	
	fileSystem		
	logicalResource		
	grupo		
	propriedades	e	e
			ou
			requirement
		ou	e
			ou
			requirement
		requirement	e
			ou
			requirement
	allocation		
	relacionamento		
	candidateResources (reservado para uso interno)	endpointReference (reservado para uso interno)	
relatedResources	logicalResource		
	grupo		
	propriedades	e	e
			ou
			requirement
		ou	e
			ou
			requirement
		requirement	e
			ou
			requirement
	allocation		
	relacionamento		
	candidateResources (reservado para uso interno)	endpointReference (reservado para uso interno)	

Tabela 111. Estrutura Hierárquica do Aqruivo JSDL (continuação)

Primeiro nível	Segundo nível	Terceiro nível	Quarto nível
optimization	objective		
	ewlm		
programando	maximumResource WaitingTime		
	estimatedDuration		
	prioridade		
	recoveryActions	action	parâmetros
			credencial
			tpmaddress
			workflow

A sintaxe JSDL usa convenções de estilo BNF para elementos e atributos:

? Indica que o elemento ou o atributo é opcional e pode ser especificado uma vez.

***** Indica que o elemento ou o atributo é opcional e pode ser especificado zero ou mais vezes.

+ Indica que o elemento ou o atributo é necessário e pode ser especificado uma ou mais vezes.

[...] Indica que os elementos ou atributos contidos entre colchetes formam um grupo.

| Indica que dois ou mais elementos ou atributos são mutuamente exclusivos.

Tipos de Elementos Job Submission Description Language

A especificação JSDL usa diversos tipos de Esquemas XML padrão. Também usa diversos tipos específicos na descrição de requisitos de tarefas.

Ambos os tipos executam uma verificação de sintaxe no valor que pode ser designado a cada elemento no arquivo JSDL. Por exemplo, o elemento **physicalMemory** segue o tipo jsdl:NumericRangeType. O tipo jsdl:NumericRangeType especifica que é possível designar a esse elemento um valor numérico específico ou um valor de intervalo numérico. Nenhum outro tipo de valor é suportado para o elemento **physicalMemory**.

Tipos de esquemas XML normativos

A especificação JSDL adota os tipos de esquemas XML (xsd) normativos listados abaixo. A sintaxe XML é um padrão de mercado e não e explicado neste manual.

- xsd:any
- xsd:anyURI
- xsd:boolean
- xsd:double
- xsd:DoubleVariableType
- xsd:duration
- xsd:IDREF
- xsd:NCName
- xsd:PriorityType
- xsd:QName

- xsd:string
- xsd:unsignedInt
- xsd:UnsignedIntVariableType

Tipos de JSDL

Os tipos a seguir são específicos da sintaxe JSDL:

StringVariableExpressionType
Um tipo de expressão de variável de sequência é um tipo simples no qual é possível especificar uma expressão de variável que pode conter uma ou mais referências de variável, como ${var}, qualquer caractere e qualquer sequência. Segue o esquema de sintaxe para esse tipo:

```
<...>
<xsd:simpleType name="StringVariableExpressionType">
  <xsd:union>
   <xsd:simpleType>
    <xsd:restriction base='xsd:string' />
   </xsd:simpleType>
   <xsd:simpleType>
    <xsd:restriction base='xsd:string'>
     <xsd:pattern
      value=".*\t*\r*\n*((\$\{[a-zA-Z_]+
          [0-9a-zA-Z_\.\-]*\})+[^\{]*[.\n]*)+" />
    </xsd:restriction>
   </xsd:simpleType>
  </xsd:union>
 </xsd:simpleType>
</...>
```

DoubleVariableExpressionType
Um tipo de expressão de variável dupla é um tipo simples no qual é possível especificar uma expressão variável que pode conter uma referência variável, como ${var}, ou um valor duplo. Segue o esquema de sintaxe para esse tipo:

```
<...>
 <xsd:simpleType name="DoubleVariableExpressionType">
  <xsd:union>
   <xsd:simpleType>
    <xsd:restriction base='xsd:double' />
   </xsd:simpleType>
   <xsd:simpleType>
    <xsd:restriction base='xsd:string'>
     <xsd:pattern value="[\n\r\t ]*($\{[a-zA-Z_]+
               [0-9a-zA-Z_\.\-]*\})[\n\r\t ]*" />
    </xsd:restriction>
   </xsd:simpleType>
  </xsd:union>
 </xsd:simpleType>

</...>
```

UnsignedIntVariableExpressionType
Um tipo de expressão de variável não assinada é um tipo simples no qual é possível especificar uma expressão variável que pode conter uma referência variável, como ${var}, ou um valor de número inteiro não assinado. Segue o esquema de sintaxe para esse tipo:

```
<...>
<xsd:simpleType name="UnsignedIntVariableExpressionType">
  <xsd:union>
   <xsd:simpleType>
    <xsd:restriction base='xsd:unsignedInt' />
```

```
   </xsd:simpleType>
   <xsd:simpleType>
    <xsd:restriction base='xsd:string'>
     <xsd:pattern value="[\n\r\t ]*($\{[a-zA-Z_]+
               [0-9a-zA-Z_\.\-]*\})[\n\r\t ]*" />
    </xsd:restriction>
   </xsd:simpleType>
  </xsd:union>
 </xsd:simpleType>

</...>
```

NotEmptyStringVariableExpressionType

Um tipo de expressão de variável de sequência é um tipo simples que permite a especificação de uma expressão de variável que pode conter uma ou mais referências de variável, como ${var}, opcionalmente em associação com qualquer caractere ou com uma sequência simples. Essa expressão de variável não pode estar vazia. Segue o esquema de sintaxe para esse tipo:

```
<xsd:simpleType name="NotEmptyStringVariableExpressionType">
  <xsd:union>
   <xsd:simpleType>
    <xsd:restriction base='xsd:string'>
     <xsd:minLength value="1"/>
    </xsd:restriction>
   </xsd:simpleType>
   <xsd:simpleType>
    <xsd:restriction base='xsd:string'>
     <xsd:pattern
      value=".*\t*\r*\n*((\$\{[a-zA-Z_]+
                    [0-9a-zA-Z_\.\-]*\})+[^\{]*[.\n]*)+" />
    </xsd:restriction>
   </xsd:simpleType>
  </xsd:union>
 </xsd:simpleType>
```

NumericRangeOnlyType

Um valor de intervalo numérico é um tipo complexo que permite a definição de intervalos e faixas maiores, menores ou contidos dentro do valor especificado. Todos os números fornecidos são expressões de variáveis duplas. Segue o esquema de sintaxe para esse tipo:

```
<...>
 <minimum>jsdl:DoubleVariableExpressionType</minimum> ?
 <maximum> jsdl:DoubleVariableExpressionType</maximum> ?
</...>
```

NumericRangeType

Um valor de intervalo numérico é um tipo complexo que permite a definição de valores ou intervalos exatos. Todos os números fornecidos são expressões de variáveis duplas. Segue o esquema de sintaxe para esse tipo:

```
<...>
 <exact>jsdl:DoubleVariableExpressionType</exact> |
 <range>jsdl:NumericRangeOnlyType</range>
</...>
```

StringRangeOnlyType

Um valor de intervalo de sequência é um tipo complexo que permite a definição de intervalos e faixas maiores, menores ou contidos dentro do valor especificado. Todos os números e sequências fornecidos são expressões de variáveis de sequência. Segue o esquema de sintaxe para esse tipo:

```
<...>
 <minimum>jsdl:StringVariableExpressionType</minimum> ?
 <maximum>jsdl:StringVariableExpressionType</maximum> ?
</...>
```

StringRangeType

Um valor de intervalo de sequência é um tipo complexo que permite a definição de valores exatos como expressões de variáveis de sequência ou intervalos que podem ser aplicados a tipos de número inteiro ou sequência. Segue o esquema de sintaxe para esse tipo:

```
<...>
 <exact>jsdl:StringVariableExpressionType</exact> |
 <range>jsdl:StringRangeOnlyType</range>
</...>
```

Elementos JSDL

O elemento principal JSDL configurado contém a semântica para elementos definidos por JSDL.

O arquivo JSDL consiste em elementos (complexos ou simples) e tipos. Elementos complexos contêm outros elementos enquanto que elementos simples não contêm nenhum outro elemento. Uma especificação de tipo executa uma verificação de sintaxe no valor especificado para o elemento ao qual se refere.

Segue uma lista dos elementos contidos na sintaxe JSDL:

Elemento jobDefinition

Definição

Esse elemento descreve a tarefa e seus requisitos. É o elemento-raiz do documento JSDL. Este atributo é requerido.

Tipo O tipo desse elemento é jsdl:JobDefinitionType. Ele pode conter os elementos a seguir:

- annotation
- category
- variables
- aplicativo
- recursos físicos
- relatedResources
- optimization
- programando

Atributos

nome O nome da tarefa especificado pelo usuário. O tipo desse atributo é **xsd:NCName**. O nome deve começar com um caractere alfabético e pode conter sublinhados (_), sinais de menos (-) e pontos (.). Espaços, caracteres especiais, caracteres acentuados e números não são suportados. Este atributo é requerido. O nome definido para esse campo identifica exclusivamente a definição de tarefa ao ser salva no banco de dados do Repositório de Tarefas. Após salvar a definição de tarefa no banco de dados, é possível enviar a definição de tarefa usando o Dynamic Workload Console ou a linha de comandos.

descrição
Uma sequência especificando uma descrição curta da definição de tarefa. O tipo desse atributo é **xsd:string**. Este atributo é opcional.

targetNamespace
Uma URI especificando o namespace de destino da definição de tarefa. O tipo desse atributo é **xsd:anyURI**. Este atributo é requerido.

Pseudo Esquema

```
<jobDefinition
 name="xsd:NCName"
 description="xsd:string"?
 xsd:anyAttribute##other>
 <annotation ... />?
 <category>.../>*
 <variables ... />?
 <application ... />
 <resources ... />?
 <relatedResources .../>*
 <optimization ...>?
 <scheduling ...>?
 <xsd:any##other/>*
</jobDefinition>
```

Elemento annotation

Definição
Esse elemento fornece informações descritivas legíveis para o ser humano sobre a definição de tarefa. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é xsd:string.

Atributos
Nenhum atributo é definido.

Pseudo Esquema

```
<annotation
 xsd:anyAttribute##other>
 xsd:string
 <xsd:any##other/>*
</application>
```

Elemento category

Definição
Esse elemento descreve a categoria da tarefa que ajuda a categorizar a tarefa. Uma tarefa pode ter diversas categorias, por exemplo: Education_DB, Financial_Dept, Asset_Management. O valor pode ser qualquer valor de sequência. Esse elemento é opcional e pode ser especificado nenhuma ou mais vezes.

Tipo O tipo desse elemento é xsd:string.

Atributos
Nenhum atributo é definido.

Pseudo Esquema

```
<category>
 xsd:string
</category>
```

Elemento variables

Definição

Esse elemento descreve a lista de variáveis definidas no arquivo JSDL. Os três tipos de variáveis a seguir são suportados:

- Cadeia
- Dupla
- Inteiro

Um valor de variável pode ser referido em outras partes do documento JSDL especificando-se: ${nome da variável}. Uma variável referida pode ser uma definida no arquivo JSDL com o elemento **variable** ou pode ser definida ao enviar a tarefa. A substituição pode ser executada pelo servidor do Dynamic Workload Broker em diferentes fases do processamento de tarefa. Em cada fase, o servidor do Dynamic Workload Broker tenta corresponder todas as referências de variáveis ainda não substituídas com variáveis definidas. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdl:VariablesType. Ele pode conter os elementos a seguir:

- stringVariable
- uintVariable
- doubleVariable

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<variables
 xsd:anyAttribute##other>
 <stringVariable ...>*
 <uintVariable ...>*
 <doubleVariable ...>*
 <xsd:any##other/>*
</variables>?
```

Elemento stringVariable

Definição

Esse elemento descreve uma variável especificando o nome da variável e o valor de sequência padrão designado. Esse elemento é opcional e pode ser especificado nenhuma ou mais vezes.

Tipo O tipo desse elemento é jsdl:StringVariableType.

Atributos

Os atributos a seguir são definidos:

nome Esse atributo especifica o nome da variável. O tipo desse atributo é xsd:NCName. Este atributo é requerido.

descrição

Esse atributo especifica a descrição da variável. O tipo desse atributo é xsd:string. Este atributo é opcional.

Pseudo Esquema

```
<stringVariable
 name="xsd:NCName"
 description="xsd:string"?
```

```
 xsd:anyAttribute##other>
 xsd:string
 <xsd:any##other/>*
</stringVariable>
```

Elemento doubleVariable

Definição

Esse elemento descreve uma variável especificando o nome da variável e o valor duplo padrão designado. Esse elemento é opcional e pode ser especificado nenhuma ou mais vezes.

Tipo O tipo desse elemento é xsd:DoubleVariableType.

Atributos

Os atributos a seguir são definidos:

nome Esse atributo especifica o nome da variável. O tipo desse atributo é xsd:NCName. Este atributo é requerido.

descrição

Esse atributo especifica a descrição da variável. O tipo desse atributo é xsd:string. Este atributo é opcional.

Pseudo Esquema

```
<doubleVariable
 name="xsd:NCName"
 description="xsd:string"?
 xsd:anyAttribute##other>
 xsd:double
 <xsd:any##other/>*
</doubleVariable>
```

Elemento uintVariable

Definição

Esse elemento descreve uma variável especificando o nome da variável e o valor de número inteiro não assinado padrão designado. Esse elemento é opcional e pode ser especificado nenhuma ou mais vezes.

Tipo O tipo desse elemento é xsd:UnsignedIntVariableType.

Atributos

Os atributos a seguir são definidos:

nome Esse atributo especifica o nome da variável. O tipo desse atributo é xsd:NCName. Este atributo é requerido.

descrição

Esse atributo especifica o nome da variável. O tipo desse atributo é xsd:string. Este atributo é opcional.

Pseudo Esquema

```
<uintVariable
 name="xsd:NCName"
 description="xsd:string"?
 xsd:anyAttribute##other>
 xsd:unsignedInt
 <xsd:any##other/>*
</uintVariable>
```

Elemento application

Definição

Esse elemento descreve o aplicativo a ser executado e os parâmetros relacionados. Esse elemento é necessário e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdl:ApplicationType.

Atributos

Os atributos a seguir são definidos:

nome Especifica o tipo do aplicativo. Os valores para os quais existe suporte são os seguintes:

Executável
Um arquivo que é utilizado para desempenhar várias funções ou operações em um computador.

J2EE Um aplicativo baseado em Java 2 Platform Enterprise Edition (J2EE).

Esse atributo é obrigatório e pode ser especificado uma vez.

descrição
Especifica o nome do aplicativo. O tipo desse atributo é xsd:string. Este atributo é opcional.

version
Especifica a versão do aplicativo. O tipo desse atributo é xsd:string. Este atributo é opcional.

Pseudo Esquema

```
<application
 name="xsd:NCName"
 description="xsd:string"?
 version="xsd:string"?
 xsd:anyAttribute##other>
 <xsd:any##other/>*
</application>
```

Elemento resources

Definição

Esse elemento contém os requisitos de recursos a tarefa que devem ser correspondidos no sistema de computador de destino para que uma tarefa seja designada a esse sistema. Os elementos de requisitos de recursos contidos são combinados em uma relacionamento AND. Isso significa que cada requisito soma aos outros para refinar o requisito do recurso de correspondência e todos os requisitos devem ser atendidos para que a alocação seja bem-sucedida. Esse elemento é opcional e pode ser especificado uma vez. Se esse elemento não estiver presente, o servidor do Dynamic Workload Broker pode escolher qualquer conjunto de recursos para executar a tarefa.

Tipo O tipo desse elemento é jsdl:ResourceType. Os tipos suportados para esse elemento estão listados na Tabela 112 na página 708. Ele pode conter os elementos a seguir:

- candidateHosts
- candidateCPUs
- physicalMemory
- virtualMemory

- candidateOperatingSystems
- fileSystem
- logicalResource
- grupo
- propriedades
- allocation
- relacionamento
- candidateResources

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<resources
 xsd:anyAttribute##other>
 <candidateHosts .../>?
 <candidateCPUs .../>?
 <physicalMemory .../>?
 <virtualMemory .../>?
 <candidateOperatingSystems .../>?
 <fileSystem .../>*
 <logicalResource ...>*
 <group ...>*
 <properties ...>
 <allocation ...>*
 <relationship ...>*
 <candidateResources ...>?
 <xsd:any##other>*
</resources>
```

Elemento relatedResources

Definição

Esse elemento é um identificador exclusivo para o requisito do recurso que deve ser exclusivo no documento. Os requisitos definidos nesse elemento se aplicam a recursos lógico e sistemas de computador. Ele pode ser referido por elementos **relationship**. Os elementos de requisitos de recursos contidos são combinados em uma relacionamento AND. Isso significa que cada requisito soma aos outros para refinar o requisito do recurso de correspondência e todos os requisitos devem ser atendidos para que a alocação seja bem-sucedida. Esse elemento é opcional e pode ser especificado nenhuma ou mais vezes.

Tipo O tipo desse elemento é jsdl:RelatedResourceType. Ele pode conter os elementos a seguir:

- logicalResource
- grupo
- propriedades
- allocation
- relacionamento
- candidateResources

Atributos

Os atributos a seguir são definidos:

id Especifica o ID interno do recurso que você deseja associar os

elemento resources. Esse ID é usado somente para referência interna dentro do arquivo JSDL. O tipo para esse atributo é xsd:ID. Este atributo é requerido.

tipo Especifica o tipo do recurso necessário. Tipos suportados são ComputerSystem e Logical Resource. Se esse atributo não estiver presente, o tipo de recurso ComputerSystem é assumido. Um tipo de recurso é identificado por um nome de tipo exclusivo e descreve as propriedades que cada instância de recurso fornece. Para obter informações adicionais sobre as propriedades de recursos disponíveis, consulte Tabela 112 na página 708. O tipo desse atributo é xsd:NCName. Este atributo é opcional.

Pseudo Esquema

```
<relatedResources
 id="xsd:ID"
 type="xsd:NCName"?
 xsd:anyAttribute##other>
 <logicalResource ...>*
 <group ...>*
 <properties ...>
 <allocation ...>*
 <relationship ...>*
 <candidateResources ...>?
 <xsd:any##other>*
</relatedResources>
```

Elemento candidateHosts

Definição

Esse elemento especifica o conjunto de hosts denominados que devem ser selecionados para execução da tarefa. O Dynamic Workload Broker designa a tarefa a um dos hosts dessa lista. Os hosts especificados estão em OR, ou seja pelo menos um deles deve ser correspondido pelo recurso Operating System contido no recurso de destino. Se nenhum dos hosts especificado estiver disponível quando a tarefa é enviada, a tarefa aguarda até que um deles esteja disponível. Se nenhum host ficar disponível antes da expiração do tempo limite, a tarefa falha. Este atributo é opcional.

Tipo O tipo desse elemento é jsdl:CandidateHostsRequirementType. Ele pode conter o elemento a seguir:

- hostName

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<candidateHosts
 xsd:anyAttribute##other>
 <hostName>jsdl:StringVariableExpressionType<hostName/>+
 <xsd:any##other>*
</candidateHosts>
```

Elemento orderedCandidatedWorkstations

Definição

Esse elemento especifica a lista ordenada de estações de trabalho que são candidatas para seleção com base nas informações especificadas no campo requirements da definição do conjunto dinâmico. A primeira estação de trabalho que corresponde aos requisitos é selecionada para processamento.

Tipo O tipo desse elemento é jsdl:OrderedCandidatedWorkstationsType. Ele pode conter o elemento a seguir:

- estação de trabalho

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<orderedCandidatedWorkstations
 xsd:anyAttribute##other>
 <workstation>jsdl:StringVariableExpressionType<workstation/>+
 <xsd:any##other>*
</orderedCandidatedWorkstations>
```

Elemento hostName

Definição

Esse elemento especifica uma expressão de variável de sequência que contém o nome de um único host que pode ser selecionado para execução da tarefa. Se nenhum dos hosts especificado estiver disponível quando a tarefa for enviada, a tarefa aguarda até que um deles esteja disponível. Se nenhum host ficar disponível antes da expiração do tempo limite, a tarefa falha. Para especificar o nome do host, é possível usar uma expressão de variável que pode conter uma ou mais referências de variáveis, como ${var}, qualquer caractere e qualquer sequência. Curingas são suportados. Esse atributo é necessário e pode ser especificado uma ou mais vezes.

Tipo O tipo desse atributo é jsdl:StringVariableExpressionType.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
 <hostName>jsdl:StringVariableExpressionType<hostName/>


 <hostName>lab145674.example.com<hostName/>

<hostName>${my_preferred_host}<hostName/>
```

Elemento candidateCPUs

Definição

Esse elemento especifica o conjunto de características da CPU que devem ser satisfeitas por hosts que podem ser selecionadas para execução da tarefa. As combinações de características especificadas estão em OR, ou seja pelo menos uma delas deve ser correspondida pelo recurso de destino. Se nenhuma das características da CPU especificadas estiver disponível quando a tarefa for enviada, a tarefa aguardará que uma delas fique disponível. Se nenhuma CPU ficar disponível antes do tempo limite vencer, a tarefa falha. Esse elemento é opcional e pode ser especificado nenhuma ou mais vezes.

Tipo O tipo desse elemento é jsdl:CandidateCPUsRequirementType. Ele pode conter o elemento a seguir:

- cpu

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<candidateCPUs
 xsd:anyAttribute##other>
 <cpu ...>*
 <xsd:any##other/>*
</candidateCPUs >?
```

Elemento cpu

Definição

Esse elemento especifica as características da CPU que devem ser satisfeitas por hosts que podem ser selecionadas para execução da tarefa. As combinações de características especificadas estão em OR, ou seja pelo menos uma delas deve ser correspondida pelo recurso de destino. Pelo menos uma das CPUs com as características especificadas deve estar disponível para que a tarefa seja executada. Se nenhuma das CPUs especificadas estiver disponível quando a tarefa for enviada, a tarefa aguarda até que uma delas esteja disponível. Se nenhuma CPU ficar disponível antes do tempo limite vencer, a tarefa falha. Esse elemento é necessário e pode ser especificado uma ou mais vezes.

Tipo O tipo desse elemento é CPURequirementType.

Atributos

Os atributos a seguir são definidos:

arquitetura

Especifica a arquitetura da CPU necessária para executar a tarefa. Este atributo é opcional. Os valores para os quais existe suporte são os seguintes:

- parisc
- powerpc
- powerpc_64
- s390
- s390x
- sparc
- x86
- x86_64

quantidade

Especifique o número de processadores que deve estar disponível no computador. Definir a Quantidade para 0 indica que o requisito é atendido por computadores com qualquer número de processadores. O tipo desse atributo é xsd:unsignedInt. Este atributo é opcional.

Pseudo Esquema

```
<cpu>
 architecture="xsd:string"?
 speed="jsdl:NumericRangeType"?
 quantity="xsd:unsignedInt"?
 xsd:anyAttribute##other>
 <xsd:any##other>*
</cpu>
```

Elemento speed

Definição

Esse elemento especifica a faixa de velocidade da CPU em MHz necessária

para executar a tarefa. A unidade de medida é megahertz. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdl:NumericRangeType.

Atributos
Nenhum atributo é definido.

Pseudo Esquema
```
<speed>
 xsd:anyAttribute##other>
 <jsdl:NumericRangeType>*
<speed>
```

Elemento physicalMemory

Definição
Esse elemento especifica uma faixa ou um valor exato que indica a quantia de memória física livre necessária para a tarefa. A quantia é expressada em bytes. Este atributo é opcional.

Tipo O tipo desse elemento é jsdl:NumericRangeType.

Atributos
Nenhum atributo é definido.

Pseudo Esquema
```
<physicalMemory
 xsd:anyAttribute##other>
 jsdl:NumericRangeType
 <xsd:any##other>*
</physicalMemory>
```

Elemento virtualMemory

Definição
Esse elemento especifica uma faixa ou um valor exato que indica a quantia de memória virtual livre necessária para a tarefa. A quantia é expressada em bytes. Este atributo é opcional.

Tipo O tipo desse elemento é jsdl:NumericRangeType.

Atributos
Nenhum atributo é definido.

Pseudo Esquema
```
<virtualMemory
 xsd:anyAttribute##other>
 jsdl:NumericRangeType
 <xsd:any##other>*
</virtualMemory>
```

Elemento candidateOperatingSystems

Definição
Esse elemento especifica o conjunto de características do sistema operacional que devem ser satisfeitas por hosts que podem ser selecionadas para execução da tarefa. As combinações de características são correspondidas em OR, ou seja pelo menos uma delas deve ser correspondida pelo recurso Operating System contido no recurso de destino. Pelo menos um dos sistemas operacionais listado deve estar disponível para a execução da tarefa. Se nenhum dos sistemas operacionais especificados estiver disponível quando a tarefa for enviada, a tarefa

aguardará que um deles fique disponível. Se nenhum ficar disponível antes da expiração do tempo limite, a tarefa falhará. Este atributo é opcional.

Tipo O tipo desse elemento é jsdl:OperatingSystemsRequirementType. Ele pode conter o elemento a seguir:

- operatingSystem

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<candidateOperatingSystems
 <xsd:anyAttribute##other>
 <operatingSystem...>*
 <xsd:any##other/>*
</ candidatOperatingSystems>?
```

Elemento operatingSystem

Definição

Esse elemento especifica as características do sistema operacional que são necessárias para execução da tarefa. Esse atributo é necessário e pode ser especificado uma ou mais vezes.

Tipo O tipo desse elemento é OperatingSystemRequirementType

Atributos

Os atributos a seguir são definidos:

tipo Esse atributo define o nome do sistema operacional necessário para execução da tarefa. O tipo desse atributo é xsd:string. Este atributo é requerido. Os valores para os quais existe suporte são os seguintes:

- AIX
- Linux
- Windows 2000
- Windows 2003
- Windows XP
- Windows Vista
- HPUX
- Solaris

version

Especifique a versão do sistema operacional. É possível especificar a versão ou sub-versão exata do sistema operacional, por exemplo, 5.2 ou 5.2.3.30, ou pode especificar uma parte da versão, por exemplo, 5.2.3. Nesse caso, o requisito se aplica a todos os sistemas operacionais que tenham a versão 5.2.3 e quaisquer sub-versões, como fix packs e níveis de manutenção. O tipo desse atributo é xsd:string. Este atributo é opcional.

Pseudo Esquema

```
<operatingSystem
 type="xsd:string"
 version="xsd:string"?
 xsd:anyAttribute##other>
 <xsd:any##other>*
</operatingSystem>
```

Elemento fileSystem

Definição

Esse elemento descreve o conjunto de características do sistema de arquivos que podem ser selecionadas para execução da tarefa. Cada conjunto de características especifica o local onde o sistema de arquivos está disponível, a quantia de espaço em disco necessária e o tipo do sistema de arquivos. O sistema de arquivos pode ser local com relação ao recurso, por exemplo, em um disco local, ou pode ser remoto, por exemplo, em um formato NFS. O requisito é satisfeito se para um determinado destino todas as características do sistema de arquivos listadas forem satisfeitas. As combinações de características são correspondidas em AND, ou seja todas devem ser correspondidas pelos recursos File System contidos no recurso de destino. Todos os sistemas de arquivos listados devem estar presentes para que a tarefa seja executada. Se qualquer um dos sistemas de arquivos especificados não estiver disponível quando a tarefa for enviada, a tarefa espera até que seja disponibilizado. Se não ficar disponível antes do tempo limite vencer, a tarefa falha. Esse elemento é opcional e pode ser especificado nenhuma ou mais vezes.

Tipo O tipo para esse elemento e jsdl:FileSystemRequirementType. Ele contém o elemento **diskSpace**.

Atributos

Os atributos a seguir são definidos:

tipo É um token que especifica o tipo de sistema de arquivos do elemento fileSystem que contém. O tipo desse atributo é jsdl:FileSystemTypeEnumeration. Este atributo é opcional. Os valores para os quais existe suporte são os seguintes:

Desconhecido
O sistema de arquivos não está especificado.

Sem Diretório Raiz
É um sistema de arquivos local diferente do diretório raiz.

Disco Removível
É um sistema de arquivos montado em um disco rígido que pode ser removido.

Disco Local
É um sistema de arquivos montado em um disco local.

Unidade Remota
É um sistema de arquivos montado em uma unidade remota.

CD-ROM
É um sistema de arquivos montado em uma unidade de CD ROM.

Disco Virtual
É um sistema de arquivos montado em um disco RAM.

mountPoint
É uma expressão de variável de sequência que especifica o mapeamento local no qual o sistema de arquivos está disponível para a tarefa. O tipo desse atributo é jsdl:StringVariableExpressionType. Para especificar o ponto de montagem, é possível usar uma expressão de variável que pode

conter uma ou mais referências de variáveis, como ${var}, qualquer caractere e qualquer sequência. Curingas são suportados. Este atributo é opcional.

Pseudo Esquema

```
<fileSystem
 type="jsdl:FileSystemTypeEnumeration"?
 mountPoint="jsdl:StringVariableExpressionType"?
 xsd:anyAttribute##other>
 <diskSpace>jsdl:NumericRangeType</diskSpace>?
 <xsd:any##other/>*
</fileSystem>
```

Elemento diskSpace

Definição

Esse elemento especifica a quantia de espaço em disco necessária no elemento do sistema de arquivos que contém para execução da tarefa. A quantia de espaço em disco é fornecida em kilobytes. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdl:NumericRangeType.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<fileSystem
 type="jsdl:FileSystemTypeEnumeration"?
 mountPoint="jsdl:StringVariableExpressionType"?
 xsd:anyAttribute##other>
 <diskSpace>jsdl:NumericRangeType</diskSpace>?
 <xsd:any##other/>*
</fileSystem>
```

Elemento logicalResource

Definição

Esse elemento especifica um ou mais recursos lógicos que são necessários para execução da tarefa. As combinações de características são correspondidas em AND, ou seja todas devem ser correspondidas pelos Logical Resources associado ao recurso de destino. Todos os recursos lógicos listados devem estar disponíveis para que a tarefa seja executada. Se um dos recursos lógicos especificados estiver indisponível quando a tarefa for enviada, a tarefa espera até que seja disponibilizado. Se não ficar disponível antes do tempo limite vencer, a tarefa falha. Esse elemento é opcional e pode ser especificado nenhuma ou mais vezes.

Tipo O tipo desse elemento é LogicalResourceRequirementType.

Atributos

Os atributos a seguir são definidos:

nome É uma expressão de variável de sequência que especifica o nome do recurso lógico solicitado. Para especificar o nome do recurso lógico, é possível usar uma expressão de variável que pode conter uma ou mais referências de variáveis, como ${var}, qualquer caractere e qualquer sequência. O nome deve começar com um caractere alfabético e pode conter sublinhados (_), sinais de menos (-) e pontos (.). Espaços, caracteres especiais e caracteres acentuados não são suportados. Este atributo é opcional.

subType
É uma expressão de variável de sequência que especifica o tipo do recurso lógico solicitado. Para especificar o tipo do recurso lógico, é possível usar uma expressão de variável que pode conter uma ou mais referências de variáveis, como ${var}, qualquer caractere e qualquer sequência. Este atributo é opcional.

quantidade
O valor de número inteiro que especifica a quantidade necessária do recurso lógico. A quantidade especificada é alocada exclusivamente para a tarefa enquanto ele é executada. Para especificar a quantia do recurso, é possível usar uma expressão de variável que pode conter uma referência de variável, como ${var}, ou um valor de número inteiro não assinado. Este atributo é opcional.

Pseudo Esquema

```
<logicalResource
 name="jsdl:StringVariableExpressionType"?
 subType="jsdl:StringVariableExpressionType"?
 quantity="jsdl:UnsignedIntVariableExpressionType"?
 xsd:anyAttribute##other>
 <xsd:any##other/>*
</logicalResource>
```

Elemento group

Definição
Esse elemento especifica que os recursos necessários para executar a tarefa precisam pertencer ao grupo de recursos especificado. Os grupos são correspondidos em AND, ou seja o recurso de destino deve estar em todos os grupos especificados. Esse elemento é opcional e pode ser especificado nenhuma ou mais vezes.

Tipo O tipo desse elemento é jsdl:GroupRequirementType.

Atributos
O atributo a seguir é definido:

nome Especifica o nome do grupo como uma expressão de variável de sequência. Para especificar o grupo de recursos, é possível usar uma expressão de variável que pode conter uma ou mais referências de variáveis, como ${var}, qualquer caractere e qualquer sequência. Esse atributo é necessário e pode ser especificado uma vez.

Pseudo Esquema

```
<group
 name="jsdl:StringVariableExpressionType"
 xsd:anyAttribute##other>
 <xsd:any##other/>*
</group>
```

Elemento properties

Definição
Esse elemento especifica as propriedades do recurso necessárias para execução da tarefa. O requisito é expressado como um conjunto de condições em propriedades de recursos combinadas com operadores AND/OR. Isso é baseado no modelo de recurso que descreve os recursos disponíveis no ambiente como instâncias de tipos de recursos. Um tipo de recurso é identificado por um nome de tipo exclusivo e descreve as

propriedades que cada instância de recurso fornece. Use esse elemento para especificar requisitos avançados. Para obter informações adicionais sobre tipos e propriedades e recursos disponíveis, consulte Tabela 112 na página 708. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdl:RequirementCompositorType. Ele pode conter os elementos a seguir:

- e
- ou
- requirement

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<properties
 xsd:anyAttribute##other>
 <and .../>?
 <or .../>?
 <requirement .../>?
 <xsd:any##other/>*
</properties>
```

Elemento and

Definição

Esse elemento especifica uma condição AND nas especificações do requisito que contém. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdl:RequirementCompositorType. Ele pode conter os elementos a seguir:

- e
- ou
- requirement

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<and
 xsd:anyAttribute##other>
 <and .../>?
 <or .../>?
 <requirement .../>?
 <xsd:any##other/>*
</and>
```

Elemento or

Definição

Esse elemento especifica uma condição OR nas especificações do requisito que contém. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdl:RequirementCompositorType. Ele pode conter os elementos a seguir:

- e
- ou
- requirement

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<or
 xsd:anyAttribute##other>
 <and .../>?
 <or .../>?
 <requirement .../>?
 <xsd:any##other/>*
</or>
```

Elemento requirement

Definição

Esse elemento é um valor de intervalo que especifica um requisito sobre os recursos de um recurso para executar a tarefa. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdl:RequirementType.

Atributos

O atributo a seguir é definido:

propertyName

É uma sequência que especifica a propriedade do recurso à qual o requisito se aplica. As propriedades disponíveis variam dependendo dos recursos selecionados no elemento recursos. Para obter informações adicionais sobre tipos e propriedades e recursos, consulte Tabela 112 na página 708. O tipo desse atributo é xsd:NCName. Este atributo é requerido.

Pseudo Esquema

```
<requirement
 propertyName="xsd:NCName"
 xsd:anyAttribute##other>
 jsdl.StringRangeType
 <xsd:any##other/>*
</and>
```

Elemento allocation

Definição

Esse elemento especifica um requisito de alocação exclusiva em uma determinada propriedade de um recurso. É possível definir um requisito de alocação nos atributos de consumo de recursos. Para obter informações adicionais sobre atributos de consumo, consulte Tabela 112 na página 708. A tarefa pode ser executada no recurso se o valor necessário estiver disponível. Enquanto a tarefa é executada, ele usa exclusivamente o valor necessário da propriedade do recurso. Quando a tarefa é concluída, ela libera o valor da propriedade. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdl:AllocationRequirementType.

Atributos

O atributo a seguir é definido:

propertyName

É uma sequência que especifica a propriedade do recurso à qual o requisito se aplica. O tipo desse atributo é xsd:QName. Este atributo é requerido.

Quantidade

Especifique a quantidade da propriedade que será alocada exclusivamente para a tarefa. Para especificar a quantidade da propriedade que deve ser alocada, é possível usar uma expressão de variável que pode conter uma referência de variável, como ${var}, ou um valor duplo.

Pseudo Esquema

```
<allocation
 propertyName="xsd:QName"
 xsd:anyAttribute##other>
 jsdl:DoubleVariableExpressionType
 <xsd:any##other/>*
</and>
```

Elemento relationship

Definição

Esse elemento especifica um requisito de que o recurso selecionado para executar a tarefa tem um relacionamento com outros recursos que correspondem a determinados critérios adicionais. Um relacionamento é uma associação direta entre um recurso de origem e um de destino. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdl:RelationshipRequirementType.

Atributos

Os atributos a seguir são definidos:

tipo É uma sequência que especifica o tipo de relacionamento necessário. O tipo desse atributo é xsd:NCName. Este atributo é opcional.

origem

É uma sequência que especifica o ID de um elemento **relatedResources**. Se esse atributo for especificado, o requisito do relacionamento especifica que o recurso deve ter pelo menos um relacionamento direcionado de um ou mais recursos correspondidos pelo elemento **relatedResources** para ele mesmo. Se esse atributo não for especificado, então um atributo de destino deve estar presente. O tipo desse atributo é xsd:IDREF. Este atributo é opcional.

target É uma sequência que especifica o ID de um elemento **relatedResources**. Se esse atributo for especificado, o requisito do relacionamento especifica que o recurso deve ter pelo menos um relacionamento direcionado dele mesmo para um ou mais recursos correspondidos pelo elemento **relatedResources**. Se esse atributo não for especificado, então um atributo de origem deve estar presente. O tipo desse atributo é xsd:IDREF. Este atributo é opcional.

Pseudo Esquema

```
<relationship
 type="xsd:NCName"
 source="xsd:IDREF"?
 target="xsd:IDREF"?
 xsd:anyAttribute##other>
 <xsd:any##other/>*
</relationship>
```

Elemento candidateResources

Esse elemento é reservado para uso interno.

Definição

Esse elemento especifica o conjunto de recursos que povem ser selecionados para execução da tarefa. Se esse elemento for especificado, um ou mais recursos do conjunto devem ser escolhido para execução da tarefa. Os recursos são identificados usando o endereço de Referência de Terminal (WS-Addressing EPR) do serviço Job Factory que está gerenciando o recurso. As combinações de requisitos são correspondidas em OR, ou seja pelo menos um deles deve ser correspondido pelo recurso contido no recurso de destino. Pelo menos um dos recursos listados deve estar disponível para a execução da tarefa. Se nenhum dos recursos especificados estiver disponível quando a tarefa for enviada, a tarefa aguarda até que um deles esteja disponível. Se nenhum rescurso ficar disponível antes do tempo limite vencer, a tarefa falha. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdl:CandidateResourcesRequirementType. Ele contém o elemento **endpointReference**.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<candidateResources
 xsd:anyAttribute##other>
 <endpointReference>wsa:EndpointReferenceType<endpointReference/>+
 <xsd:any##other>*
</candidateResources>
```

Elemento endpointReference

Esse elemento é reservado para uso interno.

Definição

Esse elemento especifica a referência de terminal de Web Services Addressing do serviço Job Factory que está gerenciando o recurso. Esse elemento é necessário e pode ser especificado uma ou mais vezes. Esse elemento é reservado para uso interno.

Tipo O tipo desse elemento é wsa:EndpointReferenceType.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<candidateResources
 xsd:anyAttribute##other>
 <endpointReference>wsa:EndpointReferenceType<endpointReference/>+
 <xsd:any##other>*
</candidateResources>
```

Elemento optimization

Definição

Esse elemento especifica as políticas de otimização a serem aplicadas à tarefa. Dependendo de se você selecionar o elemento objective ou ewlm, a política de seleção de recurso muda. Esse elemento é opcional e pode ser

especificado uma vez. Se nenhum valor for especificado, então a política de balanceamento de carga padrão para igualar o número de tarefas em execução em cada recurso é aplicada.

Tipo O tipo para esse elemento e jsdl:OptimizationType. Pode conter somente um dos seguintes elementos:

- objective
- ewlm

Atributos

O atributo a seguir é definido:

nome Especifica o nome da política de otimização. Os valores para os quais existe suporte são os seguintes:

JPT_JSDLOptimizationPolicyType
Se usar essa opção, você deve especificar o elemento **objective**. Usando o elemento **objective**, é possível especificar propriedades de recursos para serem maximizadas ou minimizadas. Ao definir o elemento **objective**, o Dynamic Workload Broker executa a tarefa no recurso que corresponde ao requisito de otimização. Este é o valor padrão. Consulte "Recursos na Definição de Tarefa" na página 707 para obter informações adicionais.

JPT_BestResource
Use essa opção para que o conjunto de recursos para a tarefa seja formado somente dos melhores recursos entre o conjunto de recursos que atendem à política especificada. Se usar essa opção, você deve especificar o elemento **objective**. Usando o elemento **objective**, é possível especificar propriedades de recursos para serem maximizadas ou minimizadas. Ao definir o elemento **objective**, o Dynamic Workload Broker executa a tarefa no recurso que corresponde ao requisito de otimização. Este é o valor padrão. Consulte "Recursos na Definição de Tarefa" na página 707 para obter informações adicionais.

JPT_EWLM
Se usar essa opção, o elemento **ewlm** é inserido automaticamente.

Pseudo Esquema

```
<optimization
 name="xsd:NCName"
 xsd:anyAttribute##other>
 <objective .../> |
 <ewlm .../>
 <xsd:any##other>*
</optimization>
```

Elemento objective

Definição

Esse elemento especifica o objetivo a ser atingido ao executar otimização para a tarefa. Por exemplo, se você selecionar **resourcePropertyName** CPU Utilization for **resourceType** Computer System with **propertyObjective** minimize, o Dynamic Workload Broker tentará executar a tarefa no recurso onde a utilização de CPU for mais baixa. Esse elemento é mutuamente exclusivo com o elemento **ewlm**.

Tipo O tipo desse elemento é PropertyObjectiveType.

Atributos

Os atributos a seguir são definidos:

propertyObjective

É uma sequência que especifica o tipo de objetivo. Este atributo é requerido. Os valores para os quais existe suporte são os seguintes:

minimize

Os recursos são atribuídos à tarefa com o objetivo de minimizar o valor da propriedade do tipo do recurso especificado. Por exemplo, é possível escolher este objetivo para atribuir a tarefa ao recurso no qual a utilização de CPU seja a mais baixa.

maximize

Os recursos são atribuídos à tarefa com o objetivo de maximizar o valor da propriedade do tipo do recurso especificado. Por exemplo, é possível escolher este objetivo para atribuir a tarefa ao recurso no qual a velocidade de processamento seja a mais alta.

minimize.utilization

Os recursos são atribuídos à tarefa com o objetivo de minimizar a utilização da propriedade do tipo do recurso especificado. Esse atributo está disponível somente para propriedades consumíveis. Para obter uma lista de todas as propriedades de consumo, consulte Tabela 112 na página 708. Se você optar por minimizar a utilização do consumo de propriedades, a tarefa será atribuída a um recurso em que seja utilizada uma quantidade menor da propriedade.

maximize.utilization

Os recursos são atribuídos à tarefa com o objetivo de maximizar a utilização da propriedade do tipo do recurso especificado. Esse atributo está disponível somente para propriedades consumíveis. Para obter uma lista de todas as propriedades de consumo, consulte Tabela 112 na página 708. Por exemplo, talvez você queira executar um teste de sobrecarga em uma estação de trabalho criando tarefas em que a propriedade de recurso Utilização da CPU para o tipo de recurso Sistema de Computador esteja definida como Maximizar Utilização. Isso faria com que todas as tarefas com essa configuração fossem designadas para a estação de trabalho onde o uso da CPU é maior, gerando um loop.

resourceType

É uma sequência que especifica o tipo do recurso ao qual a política se aplica. Se esse elemento não for especificado, o tipo de recurso ComputerSystem é assumido. O tipo desse atributo é xsd:QName. Este atributo é opcional.

resourcePropertyName

É uma sequência que especifica a propriedade do recurso à qual a política se aplica. O tipo desse atributo é xsd:QName. Este atributo é requerido.

Nota: Ao especificar um requisito de otimização em qualquer propriedade de um tipo de recurso, você deve ter definido anteriormente um requisito nesse tipo de recurso. Por exemplo, se você quiser otimizar a memória física total em um sistema operacional, deve definir previamente um requisito sobre o tipo de recurso Sistema Operacional. Esse procedimento não se aplica ao tipo de recurso Sistema de Computador, porque Sistema de Computador é o tipo de recurso padrão.

Pseudo Esquema

```
<objective
 propertyObjective="minimize" | "maximize"
 resourceType="xsd:QName"?
 resourcePropertyName="xsd:QName"
 xsd:anyAttribute##other>
 <xsd:any##other>*
</objective>
```

Elemento ewlm

Definição

Esse elemento especifica a otimização com base no cálculo do peso do recurso do Enterprise Workload Manager. O Dynamic Workload Broker executará a tarefa nos melhores recursos disponíveis, conforme indicado pelo Enterprise Workload Manager. Esse elemento é opcional e pode ser especificado uma vez. Esse elemento é mutuamente exclusivo com o elemento **objective**.

Tipo O tipo desse elemento é jsdl:PropertyObjectiveType

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<ewlm>
 xsd:anyAttribute##other>
 <xsd:any##other>*
</ewlm>
```

Elemento scheduling

Definição

Esse elemento especifica os parâmetros de planejamento a serem aplicados na execução da tarefa. Esse elemento é opcional e pode ser especificado uma vez.

No Dynamic Workload Broker, esse elemento corresponde à página Planejamento . Para obter informações adicionais sobre a página Planejamento , consulte a documentação da ajuda online.

Tipo O tipo desse elemento é jsdl:SchedulingType. Ele pode conter os elementos a seguir:

- maximumResourceWaitingTime
- estimatedDuration
- prioridade

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<scheduling
 xsd:anyAttribute##other>
 <maximumResourceWaitingTime>xsd:duration<maximumResourceWaitingTime>?
```

```
 <estimatedDuration>xsd:duration<estimatedDuration>?
 <priority>xsd:unsignedint<priority>?
 <xsd:any##other>*
</objective>
```

Elemento maximumResourceWaitingTime

Definição

Esse elemento especifica quanto tempo o servidor do Dynamic Workload Broker deve esperar desde o envio da tarefa antes de decidir que não há nenhum recurso que corresponde aos requisitos. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é xsd:duration.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<maximumResourceWaitingTime>?
 xsd:anyAttribute##other>
 <xsd:duration>*
<maximumResourceWaitingTime>
```

Elemento estimatedDuration

Definição

Esse elemento especifica a duração estimada da execução da tarefa. Isso pode ser usado pelo servidor do Dynamic Workload Broker para planejar a designação de recurso. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é xsd:duration.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<estimatedDuration>?
 <xsd:anyAttribute##other>
 <xsd:duration>*
<estimatedDuration>
```

Elemento priority

Definição

Esse elemento especifica a prioridade da tarefa como um número inteiro entre 0 e 100. Valores mais altos significam prioridade mais alta. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é xsd:PriorityType. Ele pode conter os elementos a seguir:

- maximumResourceWaitingTime
- estimatedDuration
- prioridade

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<scheduling
 xsd:anyAttribute##other>
 <maximumResourceWaitingTime>xsd:duration<maximumResourceWaitingTime>?
```

```
 <estimatedDuration>xsd:duration<estimatedDuration>?
 <priority>xsd:unsignedint<priority>?
 <xsd:any##other>*
</objective>
```

recoveryActions

Definição

Esse elemento descreve a lista de ações de recuperação que o Dynamic Workload Broker deve executar se o intervalo de tempo especificado no elemento maximumResourceWaitingTime vencer e nenhum recurso que corresponda aos requisitos tiver sido localizado. Esse elemento é opcional e pode ser especificado uma vez. As ações de recuperação definidas nesse elemento são executadas iniciando um fluxo de trabalho do Tivoli Provisioning Manager.

Tipo O tipo desse elemento é jsdl:RecoveryActionList. Ele pode conter o elemento de ação.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<recoveryActions
 xsd:anyAttribute##other>
 <action...>+
 <xsd:any##other/>*
</recoveryActions >
```

action

Definição

Esse elemento especifica a(s) ação(ões) a ser(em) executada(s). O Dynamic Workload Broker executa as ações listadas sequencialmente com base na ordem na qual foram especificadas. Qualquer ação de recuperação subsequente é executada somente se a ação anterior tiver sido concluída com sucesso. Se o elemento **recoveryActions** for especificado, pelo menos um elemento **action** deve ser especificado.

Tipo O tipo desse elemento é jsdl:RecoveryActionType.

Atributos

Os atributos a seguir são definidos:

nome Especifica o nome da ação de recuperação a ser executada. Este atributo é opcional.

additionalTimeOnCompletion

Especifica o intervalo de tempo que o Dynamic Workload Broker deve esperar até a ação de recuperação se tornar efetiva após a conclusão. Se esse atributo for especificado para uma ação de recuperação, a ação de recuperação subsequente é executada somente após o intervalo de tempo especificado vencer. Se o recurso necessário ficar disponível antes do intervalo vencer, o Dynamic Workload Broker pode decidir executar a tarefa antes da ação ser concluída.

maximumExecutionTime

Especifica o período de tempo esperado que o Dynamic Workload Broker espera até a ação de recuperação ser concluída. Se a ação de

recuperação não estiver concluída quando o tempo limite vencer, o procedimento de recuperação falha e a sequência de recuperação é parada.

Pseudo Esquema

```
< action>
 name="xsd:NCName"
 additionalTimeOnCompletion ="xsd:duration"?
 maximumExecutionTime ="xsd:duration"?
 xsd:anyAttribute##other>
 <xsd:any##other/>*
</ action >
```

Elemento tpmaction

Definição

Esse elemento especifica os parâmetros necessários para executar uma ação de recuperação do Tivoli Provisioning Manager. Esse atributo é opcional e pode ser especificado uma vez.

Tipo O tipo desse atributo é jsdltpm:TPMActionType. Ele pode conter os elementos a seguir:

- parâmetros
- credencial
- tpmaddress
- workFlow

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<tpmaction
 <parameters... />?
 <credential.../>?
 <tpmaddress.../>?
 workFlow="jsdl:StringVariableExpressionType"
</tpmaction>
```

Elemento parameters

Definição

Esse elemento especifica os argumentos a serem usados ao executar o fluxo de trabalho Tivoli Provisioning Manager. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse atributo é jsdltpm:ParametersType.

Atributos

Os atributos a seguir são suportados:

nome Especifica o nome do atributo a ser usado ao executar o fluxo de trabalho do Tivoli Provisioning Manager. Esse atributo é necessário e pode ser especificado uma ou mais vezes.

Pseudo Esquema

```
<parameters
 <parameter... />+
</parameters>
```

Elemento credential

Definição

Esse elemento especifica as credenciais necessárias para executar o fluxo de trabalho do Tivoli Provisioning Manager. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdl:CredentialType. Ele pode conter os elementos a seguir:

- userName
- password

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<credential
 <userName> jsdl:NotEmptyStringVariableExpressionType</userName>
 <password> jsdl:StringVariableExpressionType </password>
</credential>
```

Elemento userName

Definição

Esse elemento especifica um nome de usuário definido no sistema de destino que é usado para executar o fluxo de trabalho do Tivoli Provisioning Manager. Esse elemento é necessário se você usar o elemento credential e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdl:NotEmptyStringVariableExpressionType.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<userName>
 xsd:anyAttribute##other>
 <jsdl:NotEmptyStringVariableExpressionType*
<userName>
```

Elemento password

Definição

Esse elemento especifica a senha do usuário especificado que é usado para executar o fluxo de trabalho do Tivoli Provisioning Manager no sistema de destino. Esse elemento é necessário se você usar o elemento credential e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdl:StringVariableExpressionType.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<password>?
 xsd:anyAttribute##other>
 <jsdl:StringVariableExpressionType
<password>
```

Elemento tpmaddress

Definição

Esse elemento especifica o endereço do Tivoli Provisioning Manager que deve ser usado para chamar o serviço da Web do Tivoli Provisioning

Manager necessário para executar o fluxo de trabalho. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdltpm:TPMAddressType.

Atributos

Os atributos a seguir são definidos:

host Especifica o nome do host do servidor do Tivoli Provisioning Manager a ser usado ao executar a ação de recuperação.

porta Especifica o número da porta do servidor do Tivoli Provisioning Manager a ser usado ao executar a ação de recuperação.

Pseudo Esquema

```
< tpmaddress
 <host... />
 <port... />
</ tpmaddress >
```

Elemento workflow

Definição

Esse elemento especifica o nome do fluxo de trabalho do Tivoli Provisioning Manager a ser executado. Para especificar o nome do fluxo de trabalho, é possível usar uma expressão de variável que pode conter uma ou mais referências de variáveis, como ${var}, qualquer caractere e qualquer sequência. Esse elemento é necessário.

Tipo O tipo desse elemento é jsdl:StringVariableExpressionType.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<workflow>?
 xsd:anyAttribute##other>
 <jsdl:StringVariableExpressionType
<workflow>
```

Elemento executable

Definição

Esse elemento especifica os parâmetros para executar um comando do sistema nativo, ou seja arquivos executáveis e scripts. Também é possível integrar scripts nesse elemento. Esse elemento é necessário.

Nota: As seguintes restrições se aplicam:

- Nos sistemas Windows, é possível executar scripts que contêm comandos em lote. Os formatos suportados para scripts são:
  - .cmd
  - .bat
- Nos sistemas UNIX e Linux, somente scripts shell são suportados. No início do script da shell, especifique o interpretador de comandos.
- Em sistemas UNIX e Linux, comandos contidos em scripts devem ser executado em primeiro plano. Isto significa que você não pode utilizar o parâmetro "&" em conjunto com o comando.
- Em todas as plataformas suportadas, você não pode incluir nas tarefas qualquer comando iniciando com uma interface gráfica.

Tipo O tipo desse elemento é jsdle:ExecutableType. Ele pode conter os elementos a seguir:

- script
- argumentos
- environment
- credencial

Atributos

Os atributos a seguir são definidos:

path É uma expressão de variável de sequência que especifica o nome do caminho do arquivo executável a ser executado. Se o elemento **script** não estiver presente, o atributo **path** deve ser especificado. Se o elemento **script** estiver presente, o atributo **path** não pode ser especificado. Para especificar o caminho, é possível usar uma expressão de variável que pode conter uma ou mais referências de variáveis, como ${var}, qualquer caractere e qualquer sequência.

É necessário especificar a extensão do arquivo. Se desejar executar um arquivo executável sem especificar sua extensão, é possível especificar o nome do arquivo executável no elemento **script**, de forma que o arquivo seja executado sem o shell.

input É uma expressão de variável de sequência que especifica a entrada padrão para o comando. Esse atributo é um nome absoluto do caminho ou um nome do caminho relativo ao diretório ativo. Para especificar o caminho, é possível usar uma expressão de variável que pode conter uma ou mais referências de variáveis, como ${var}, qualquer caractere e qualquer sequência. Este atributo é opcional.

output

É uma expressão de variável de sequência que especifica a saída padrão para o comando. Esse atributo é um nome absoluto do caminho ou um nome do caminho relativo ao diretório ativo. Para especificar o caminho, é possível usar uma expressão de variável que pode conter uma ou mais referências de variáveis, como ${var}, qualquer caractere e qualquer sequência. Este atributo é opcional.

erro É uma expressão de variável de sequência que especifica o erro padrão para o comando. Esse atributo é um nome absoluto do caminho ou um nome do caminho relativo ao diretório ativo. Para especificar o caminho, é possível usar uma expressão de variável que pode conter uma ou mais referências de variáveis, como ${var}, qualquer caractere e qualquer sequência. Este atributo é opcional.

workingDirectory

É uma expressão de variável de sequência que especifica o diretório ativo necessário para a tarefa a ser executada. Para especificar o diretório, é possível usar uma expressão de variável que pode conter uma ou mais referências de variáveis, como ${var}, qualquer caractere e qualquer sequência. Este atributo é opcional. Se esse atributo não do especificado, a tarefa é executada nos diretórios a seguir, dependendo do sistema operacional:

Em sistemas UNIX

Os seguintes casos são aplicáveis:

- A tarefa é executada em $HOME_DIRECTORY do usuário que envia a tarefa, se existir.
- Se este diretório não existir, ele é executado em /root, se o usuário que envia a tarefa tiver os direitos necessários.
- Se o usuário não tiver os direitos necessários, a tarefa é executada no diretório ativo do Agente do Tivoli Workload Scheduler.

Em sistemas Windows
A tarefa é executada no diretório ativo do Agente do Tivoli Workload Scheduler.

script Especifica o código de script a ser executado. Para especificar caracteres especiais necessários para linguagens de script, o conteúdo do elemento de script pode ser especificado com uma seção CDATA

Pseudo Esquema

```
<executable
 path="jsdl:StringVariableExpressionType"
 input="jsdl:StringVariableExpressionType"?
 output="jsdl:StringVariableExpressionType"?
 error="jsdl:StringVariableExpressionType"?
 workingDirectory="jsdl:StringVariableExpressionType"?
 xsd:anyAttribute##other>
 <script ... />?
 <arguments .../>?
 <environment .../>?
 <credential .../>?
 <xsd:any##other>*
</executable>
```

Elemento de script

Definição
Esse elemento especifica o código de script a ser executado. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é xsd:string.

Atributos
Nenhum atributo é definido.

Pseudo Esquema

```
<script>?
 xsd:anyAttribute##other>
 <xsd:string
<script>
```

Elemento arguments

Definição
Esse elemento especifica a lista de argumentos como expressões de variáveis de sequência que são concatenadas para produzir a sequência de argumentos a ser passada ao comando. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdle:ArgumentsType. Ele pode conter o elemento a seguir:

- valor

Atributos
Nenhum atributo é definido.

Pseudo Esquema

```
<arguments
 xsd:anyAttribute##other>
 <value>jsdl:StringVariableExpressionType</value>+
 <xsd:any##other>*
</arguments>
```

Elemento value

Definição

Esse elemento especifica o valor do elemento **arguments**. Para especificar o valor, é possível usar uma expressão de variável que pode conter uma ou mais referências de variáveis, como ${var}, qualquer caractere e qualquer sequência. Esse elemento é necessário e pode ser especificado uma ou mais vezes.

Tipo O tipo desse elemento é jsdl:StringVariableExpressionType.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<arguments
 xsd:anyAttribute##other>
 <value>jsdl:StringVariableExpressionType</value>+
 <xsd:any##other>*
</arguments>
```

Nota: Se for necessário especificar que um valor consiste em um espaço em branco, você deve colocá-lo entre aspas duplas.

Elemento environment

Definição

Esse elemento especifica uma expressão de variável de sequência de variáveis de ambiente que serão definidas para a tarefa no ambiente de execução. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdl:jsdle:EnvironmentType. Ele pode conter o elemento a seguir:

- variável

Atributos

Nenhum atributo definido

Pseudo Esquema

```
<environment
 xsd:anyAttribute##other>
 <variable name="xsd:string">jsdl:StringVariableExpressionType</variable>+
 <xsd:any##other>*
</environment>
```

Nota: Se for necessário especificar que um valor consiste em um espaço em branco, você deve colocá-lo entre aspas duplas.

Elemento variable

Definição

Esse elemento especifica uma expressão de variável de sequência de variáveis de ambiente que serão definidas para a tarefa no ambiente de execução. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdl:StringVariableExpressionType.

Atributos

Os atributos a seguir são definidos:

nome Especifica o nome da variável.

valor Especifica o valor da variável. Para especificar o valor da variável, é possível usar uma expressão de variável que pode conter uma ou mais referências de variáveis, como ${var}, qualquer caractere e qualquer sequência.

Elemento credential

Definição

Esse elemento especifica a credencial de segurança para executar o comando. Inclua esse elemento quando quiser especificar um usuário ou nome do grupo sob o qual o executável ou o script é executado no sistema de destino que é diferente do usuário ou nome do grupo sob o qual o agente de carga de trabalho é executado. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdle:CredentialType. Ele pode conter os elementos a seguir:

- userName
- groupName
- password

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<credential
 xsd:anyAttribute##other>
 <userName> jsdl:StringVariableExpressionType </userName>
 <groupName> jsdl:StringVariableExpressionType </groupName>
 <password> jsdl:StringVariableExpressionType </password>
 <xsd:any##other>*
</credential>
```

Elemento userName

Definição

É uma expressão de variável de sequência que especifica o nome de usuário de um usuário definido no sistema de destino. O comando executa utilizando este nome de usuário. Esse elemento é necessário se você usar o elemento credential e pode ser especificado uma vez. Pode ser um ID do usuário do UNIX ou do Windows. Para especificar o nome de usuário, é possível usar uma expressão de variável que pode conter uma ou mais referências de variáveis, como ${var}, opcionalmente, em associação com qualquer caractere ou com uma sequência simples. Se o aplicativo for executado em um sistema Windows como um usuário do domínio do Windows, especifique o nome de usuário conforme a seguir:

```
domain_name\user_name
```

Se o aplicativo executar como um usuário local, é possível utilizar o seguinte formato:

```
user_name
```

Tipo O tipo desse elemento é jsdl:NotEmptyStringVariableExpressionType.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<credential
 xsd:anyAttribute##other>
 <userName> jsdl:StringVariableExpressionType </userName>
 <groupName> jsdl:StringVariableExpressionType </groupName>
 <password> jsdl:StringVariableExpressionType </password>
 <xsd:any##other>*
</credential>
```

Elemento groupName

Definição

É uma expressão de variável de sequência que especifica o nome do grupo ao qual o usuário pertence que está definido no sistema de destino onde o comando é executado. Esse elemento é opcional e pode ser especificado uma vez. Esse elemento é ignorado em sistemas de destino do Windows. Para especificar o nome do grupo, é possível usar uma expressão de variável que pode conter uma ou mais referências de variáveis, como ${var}, qualquer caractere e qualquer sequência.

Tipo O tipo desse elemento é jsdl:StringVariableExpressionType.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<credential
 xsd:anyAttribute##other>
 <userName> jsdl:StringVariableExpressionType </userName>
 <groupName> jsdl:StringVariableExpressionType </groupName>
 <password> jsdl:StringVariableExpressionType </password>
 <xsd:any##other>*
</credential>
```

Elemento password

Definição

É uma expressão de variável de sequência que define a senha do nome de usuário especificado que é usado para executar o comando no sistema de destino. Esse elemento é opcional e pode ser especificado uma vez. Esse elemento é ignorado em sistemas de destino do UNIX. Para especificar a senha, é possível usar uma expressão de variável que pode conter uma ou mais referências de variáveis, como ${var}, qualquer caractere e qualquer sequência.

Tipo O tipo desse elemento é jsdl:StringVariableExpressionType.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<credential
 xsd:anyAttribute##other>
 <userName> jsdl:StringVariableExpressionType </userName>
 <groupName> jsdl:StringVariableExpressionType </groupName>
 <password> jsdl:StringVariableExpressionType </password>
 <xsd:any##other>*
</credential>
```

Elemento j2ee

Definição

Esse elemento especifica as informações do aplicativo J2EE necessárias para a tarefa. Esse elemento é opcional e pode ser especificado uma vez. As operações J2EE que é possível executar variam conforme o tipo de planejador (direto ou indireto) selecionado e se a segurança do WebSphere Application Server ou J2EE é ativada ou não.

Tipo O tipo desse elemento é jsdlj:J2EEType. Ele pode conter os elementos a seguir:

- invoker
- jsm
- ejb
- credencial

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<j2ee>?
 xsd:anyAttribute##other>
 <jsdlj:J2EEType
<j2ee>
```

Elemento invoker

Definição

Esse elemento especifica se o invocador indireto ou direto deve ser usado para o aplicativo J2EE. Esse elemento é necessário e pode ser especificado uma vez. Selecionar um invocador direto significa que o Agente do Tivoli Workload Scheduler encaminha a tarefa imediatamente para os componentes da instância do WebSphere Application Server (EJB ou JMS). Selecionar um invocador indireto significa que o Agente do Tivoli Workload Scheduler usa uma infraestrutura de planejamento do WebSphere existente já configurada no WebSphere Application Server de destino.

Tipo O tipo desse elemento é jsdlj:InvokerType.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<invoker>?
 xsd:anyAttribute##other>
 <jsdlj:InvokerType
<invoker>
```

Elemento jms

Definição

Esse elemento especifica a fila de destino do Java Message System (JMS) e a mensagem a ser enviada. Esse elemento é opcional e pode ser especificado uma vez. É mutuamente exclusivo com o elemento **ejb**.

Tipo O tipo desse elemento é jsdlj:JMSActionType.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<jms>?
 xsd:anyAttribute##other>
 <jsdlj:JMSActionType
<jms>
```

Elemento ejb

Definição

Esse elemento especifica as características do início de JNDI do EJB a ser chamado. É mutuamente exclusivo com o elemento **jms**. O EJB já deve estar instalado no WAS de destino e deve implementar a interface TaskHandler.

Tipo O tipo desse elemento é jsdlj:EJBActionType. Ele pode conter os elementos a seguir:

- jndiHome
- credencial

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<ejb>?
 xsd:anyAttribute##other>
 <jsdlj:EJBActionType
<ejb>
```

Elemento jndiHome

Definição

Esse elemento especifica o diretório inicial da interface de programação de aplicativos Java Naming and Directory Interface (JNDI). Esse elemento é necessário e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdl:StringVariableExpressionType.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<jndiHome>?
 xsd:anyAttribute##other>
 <jsdl:StringVariableExpressionType
<jndiHome>
```

Elemento ejb

Definição

Esse elemento especifica as características da ação do JMS.

Tipo O tipo desse elemento é jsdlj:JMSActionType. Ele pode conter os elementos a seguir:

- connFactory
- destino
- message
- credencial

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<jms>?
 xsd:anyAttribute##other>
 <jsdl:JMSActionType
<jms>
```

Elemento connFactory

Definição

Esse elemento especifica um objeto administrado que um cliente usa para criar uma conexão com o provedor JMS. Esse elemento é necessário e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdl:StringVariableExpressionType.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<connFactory>
 xsd:anyAttribute##other>
 <jsdl:StringVariableExpressionType
<connFactory>
```

Elemento destination

Definição

Esse elemento especifica um objeto administrado que encapsula a identidade de um destino de mensagem, que é onde as mensagens são entregues e consumidas. Esse elemento é necessário e pode ser especificado uma vez.

Multiplicidade

Tipo O tipo desse elemento é jsdl:StringVariableExpressionType.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

Elemento message

Definição

Esse elemento especifica um objeto que é enviado de um aplicativo para outro. Esse elemento é necessário e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdl:StringVariableExpressionType.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<message>
 xsd:anyAttribute##other>
 <jsdl:StringVariableExpressionType
<message>
```

Elemento credential

Definição

Esse elemento especifica as credenciais necessárias para executar o aplicativo J2EE. Inclua esse elemento quando quiser especificar um nome de usuário sob o qual o aplicativo é executado no sistema de destino que é

diferente do nome de usuário sob o qual o agente de carga de trabalho é executado. Esse elemento é opcional e pode ser especificado uma vez.

Tipo O tipo desse elemento é jsdl:CredentialType. Ele pode conter os elementos a seguir:

- userName
- password
- JAASAuthenticationAlias

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<credential>?
 xsd:anyAttribute##other>
 <jsdl:CredentialType
<credential>
```

Elemento userName

Definição

Esse elemento especifica o nome de usuário de um usuário definido no sistema de destino. O aplicativo J2EE é executado usando esse nome de usuário. Esse elemento é necessário se você usar o elemento credential e pode ser especificado uma vez. Para especificar o nome de usuário, é possível usar uma expressão de variável que pode conter uma ou mais referências de variáveis, como ${var}, opcionalmente em associação com qualquer caractere ou com uma sequência simples. Se você escolher um invocador indireto, use esse elemento para especificar o nome de usuário necessário para conectar ao planejador do WebSphere Application Server.

Tipo O tipo desse elemento é jsdl:NotEmptyStringVariableExpressionType.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<userName>
 xsd:anyAttribute##other>
 <jsdl:NotEmptyStringVariableExpressionType
<userName>
```

Elemento password

Definição

Esse elemento especifica a senha do nome de usuário especificado que é usado para executar o aplicativo J2EE no sistema de destino. Esse elemento é opcional e pode ser especificado uma vez. Para especificar a senha, é possível usar uma expressão de variável que pode conter uma ou mais referências de variáveis, como ${var}, qualquer caractere e qualquer sequência. Se você escolher um invocador indireto, use esse elemento para especificar a senha necessária para conectar ao planejador do WebSphere Application Server.

Tipo O tipo desse elemento é jsdl:StringVariableExpressionType.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<password>
 xsd:anyAttribute##other>
 <jsdl:StringVariableExpressionType
<password>
```

Elemento JAASAuthenticationAlias

Definição

Esse elemento especifica o alias de autenticação de JAAS. Esse elemento é opcional e pode ser especificado uma vez. É necessário somente ao usar um invocador indireto. Para especificar o alias, é possível usar uma expressão de variável que pode conter uma ou mais referências de variáveis, como ${var}, opcionalmente em associação com qualquer caractere ou com uma sequência simples.

Tipo O tipo desse elemento é jsdl:StringVariableExpressionType.

Atributos

Nenhum atributo é definido.

Pseudo Esquema

```
<JAASAuthenticationAlias>
 xsd:anyAttribute##other>
 <jsdl:NotEmptyStringVariableExpressionType
<JAASAuthenticationAlias>
```

Recursos na Definição de Tarefa

Este tópico fornece uma visão geral de como os recursos e suas propriedades são utilizados na definição de tarefa para identificar os possíveis destinos, para reservar as alocações de recursos consumíveis e para otimizar o balanceamento de carga entre os recursos disponíveis.

Um entendimento dos recursos físicos e lógicos e suas propriedades é a chave para criar uma definição de tarefa que focaliza exatamente os recursos adequados para executar a tarefa, determina o requisito de alocação de recursos e contribui para o balanceamento da carga entre os recursos disponíveis. Cada recurso possui uma ou mais propriedades associadas a ele. As propriedades podem ter as seguintes características:

É consumível

As propriedades de recursos que são consumíveis possuem uma quantidade finita associada a eles que podem ser consumidas pelas tarefas que são alocadas para o recurso. Por exemplo, um sistema de computador possui um número finito de processadores.

Pode ser otimizado

Algumas propriedades podem ser usadas para definir a otimização de objetivos, que determina como a carga deve ser equilibrada quando as tarefas forem alocadas para um grupo de recursos. Por exemplo, é possível escolher alocar uma tarefa para o recurso correspondente que possua uso de CPU mais baixo.

Suporta Caracteres Curinga

Algumas propriedades podem ser especificadas na definição de tarefa usando curingas. Por exemplo, um requisito para uma determinada série de modelos de computadores pode ser definido ao especificar o modelo usando curingas.

A Tabela 112 mostra os diferentes tipos de recursos que podem ser incluídos em uma definição de tarefa e suas propriedades disponíveis.

Tabela 112. Tipos e Propriedades de Recursos

Tipo de Recurso	Propriedades Disponíveis	É consumível	Pode ser otimizado	Suporta Caracteres Curinga
ComputerSystem	CPUUtilization	Não	Sim	Não
	HostName	Não	Não	Sim
	Fabricante	Não	Não	Sim
	Modelo	Não	Não	Sim
	NumOfProcessors	Sim	Sim	Não
	ProcessingSpeed	Não	Sim	Não
	ProcessorType	Não	Não	Não
LogicalResource	DisplayName	Não	Não	Sim
	SubType	Não	Não	Sim
	Quantidade	Sim	Sim	Não
OperatingSystem	DisplayName	Não	Não	Sim
	FreePhysicalMemory	Não	Sim	Não
	FreeSwapSpace	Não	Sim	Não
	FreeVirtualMemory	Não	Sim	Não
	OperatingSystemType	Não	Não	Não
	OperatingSystem Versão	Não	Não	Não
	TotalPhysicalMemory	Sim	Sim	Não
	TotalSwapSpace	Sim	Sim	Não
	TotalVirtualMemory	Sim	Sim	Não
FileSystem	DisplayName	Não	Não	Sim
	FileSystemRoot	Não	Não	Sim
	FileSystemType	Não	Não	Não
	FreeStorageCapacity	Não	Sim	Não
	TotalStorageCapacity	Sim	Sim	Não
NetworkSystem	NetworkAddress	Não	Não	Não
	NetworkSystem HostName	Não	Não	Sim

Apêndice C. Referência rápida para comandos

Este apêndice está dividido em quatro seções:

- "Gerenciando o Plano"
- "Gerenciando objetos no banco de dados" na página 710
- "Gerenciando objetos no plano" na página 718
- "Comandos utilitários" na página 722
- "Comandos de relatório" na página 724

Gerenciando o Plano

Esta seção descreve as operações que podem ser executadas contra o plano utilizando o script **JnextPlan** e a linha de comandos **planman**:

Tabela 113. Comandos Utilizados Contra o Plano

Sintaxe do Comando ou do Script	Ação Executada
JnextPlan [**-from** *mm/dd/[yy]yy[hh[:]mm[**tz** | **timezone** tzname]]*] {**-to** *mm/dd/[yy]yy[hh[:]mm[**tz** | **timezone** tzname]]* | **-for** *[h]hh[:]mm* [**-days** *n*] | **-days** *n*}	Cria ou estende o plano de produção.
planman [*connection_parameters*] **crt** [**-from** *mm/dd/[yy]yy* [*hh[:]mm* [**tz** | **timezone** *tzname*]]] {**-to** *mm/dd/[yy]yy[hh[:]mm[**tz** | **timezone** tzname]]* | **-for** *[h]hh[:]mm* [**-days** *n*] | **-days** *n*}	Cria um plano de produção intermediário.
planman [*connection_parameters*] **deploy** [**-scratch**]	Implementa todas as regras que não estejam no estado de rascunho.
planman [*connection_parameters*] **ext** {**-to** *mm/dd/[yy]yy[hh[:]mm[**tz** | **timezone** tzname]]* | **-for** *[h]hh[:]mm* [**-days** *n*] | **-days** *n*}	Cria um plano intermediário para uma extensão de plano.
planman [*connection_parameters*] **showinfo**	Recupera as informações do plano de produção.
planman [*connection_parameters*] **crttrial** *file_name* [**-from** *mm/dd/[yy]yy* [*hh[:]mm* [**tz** | **timezone** *tzname*]]] {**-to** *mm/dd/[yy]yy[hh[:]mm[**tz** | **timezone** tzname]]* | **-for** *[h]hh[:]mm* [**-days** *n*] | **-days** *n*}	Cria um plano trial.
planman [*connection_parameters*] **exttrial** *file_name* {**-to** *mm/dd/[yy]yy[hh[:]mm[**tz** | **timezone** tzname]]* | **-for** *[h]hh[:]mm* [**-days** *n*] | **-days** *n*}	Cria um plano trial de uma extensão de plano de produção.
planman [*connection_parameters*] **crtfc** *file_name* [**-from** *mm/dd/[yy]yy* [*hhmm* [**tz** | **timezone** *tzname*]]] {**-to** *mm/dd/[yy]yy[hh[:]mm[**tz** | **timezone** tzname]]* | **-for** *[h]hh[:]mm* [**-days** *n*] | **-days** *n*}	Cria um plano de previsão.
planman [*connection_parameters*] **unlock**	Desbloqueia o plano de produção.

© Copyright IBM Corp. 1999, 2014

Tabela 113. Comandos Utilizados Contra o Plano (continuação)

Sintaxe do Comando ou do Script	Ação Executada
ResetPlan [*connection_parameters*] [**-scratch**]	Reconfigura o plano de produção.
planman reset -scratch	Remove o plano de pré-produção enquanto mantém o arquivo do Symphony.
planman [*connection_parameters*] **resync**	Replica os dados do plano a partir do arquivo do Symphony no banco de dados.
planman [*connection_parameters*] **checksync**	Monitora o progresso e o resultado do processo de replicação dos dados do plano no banco de dados.

em que *connection_parameters* são o seguinte:

```
[-file filename]
[-host hostname]
[-port port_name]
[-protocol protocol_name][-proxy proxy_name]
[-proxyport proxy_port_number]
[-password user_password]
[-timeout seconds]
[-username user_name]
```

Para obter informações adicionais, consulte a "Criando e estendendo o plano de produção" na página 79.

Gerenciando objetos no banco de dados

A seção está dividida nas seguintes subseções:

- "Comandos de Objetivo Geral"
- "Objetos de Planejamento"
- "Comandos do Composer" na página 715

Comandos de Objetivo Geral

Esta seção descreve os nomes, a sintaxe dos comandos de objetivo geral que são executados a partir do programa **composer** e a autorização do usuário, quando necessária, para sua execução.

Tabela 114. Comandos de Objetivo Geral

Comando	Sintaxe	Autorização do Usuário
continuar	**continue**&*command argument*&*command argument*	Autorização para utilizar o composer
edit	**edit** *filename*	Autorização para utilizar o composer
exit	**exit**	Autorização para utilizar o composer
help	**help** *commandname*	Autorização para utilizar o composer
redo	**redo** *directives*	Autorização para utilizar o composer
validate	**validate** *filename* [;syntax]	Autorização para utilizar o composer
version	**versão**	Autorização para utilizar o composer

Objetos de Planejamento

Esta seção contém a sintaxe de definição de todos os objetos de planejamento.

Na tabela que exibe a lista de comandos que podem ser utilizados com o objeto de planejamento, *filename* indica um arquivo existente quando utilizado na sintaxe para os comandos **add** e **replace**, indica um arquivo não existente quando utilizado na sintaxe para o comando **create/extract**.

Calendário

Sintaxe de definição do arquivo:

$calendar
calendarname [*"description"*]
date [...]

Domínio

Sintaxe de definição do arquivo:

domain *domainname*[**description** *"description"*]
*** manager** *workstation*
[**parent** *domainname* | **ismaster**]
end

Regra de Evento

Sintaxe de definição XML:

- eventRule name=" " ruleType=" " isDraft=" " (1, 1)
  - description (0, 1)
  - timeZone (0, 1)
  - validity from=" " to=" " (0, 1)
  - activeTime start=" " end=" " (0, 1)
  - timeInterval amount=" " unit=" " (0, 1)
  - eventCondition eventProvider=" " eventType=" " (1, n)
    - scope (0, 1)
    - filteringPredicate (0, 1)
      - attributeFilter name=" " operator="eq" (0, n)
        - value (1, n)
      - attributeFilter name=" " operator="ne" (0, n)
        - value (1, n)
      - attributeFilter name=" " operator="le" (0, n)
        - value (1, 1)
      - attributeFilter name=" " operator="ge" (0, n)
        - value (1, 1)
      - attributeFilter name=" " operator="range" (0, 1)
        - value (1, 2)
  - correlationAttributes (0, 1)
    - attribute name=" " (1, n)
  - action actionProvider=" " actionType=" " responseType=" " (0, n)
    - description (0, 1)
    - scope (0, 1)
    - parameter name=" "(1, n)
    - value (1, 1)

Tarefa

Sintaxe de definição do arquivo:

$jobs
[*workstation*#]*jobname*
{**scriptname** *filename* | **docommand** *"command"* | **task** *job_definition*}
streamlogon *username*
[**description** *"description"*]
[**tasktype** *tasktype*]
[**interactive**][1]
[**rccondsucc** "*Success Condition*"]

[**recovery**
{**stop** | **continue** | **rerun**}
[**after** [*workstation*#]*jobname*]
[**abendprompt** *"text"*]]

Nota:

1. Esta palavra-chave está disponível somente na plataforma Windows.

Fluxo de Tarefas

Sintaxe de definição do arquivo:

schedule [*workstation*#]*jobstreamname*
comment
[**validfrom** *date*]
[**timezone**|**tz** *tzname*]
[**description** *"text"*]
[**draft**]
[**vartable** *table_name*]
[**freedays** *calendarname* [**-sa**] [**-su**]]
[**on** [**runcycle** *name*]
[**validfrom** *date*] [**validto** *date*]
[**description** *"text"*]
[**vartable** *table_name*]
{*date*|*day*|*calendar*|*request*|*"icalendar"*|*runcyclegroup*} [**,...**]
[**fdignore**|**fdnext**|**fdprev**]
[({**at** *time* [+*n* **day**[**s**]] |
schedtime *time* [+*n* **day**[**s**]]}
[**until** *time* [+*n* **day**[**s**]] [**onuntil** *action*]]
[**deadline** *time* [+*n* **day**[**s**]]])]]
[**,...**]
[**except** [**runcycle** *name*]
[**validfrom** *date*] [**validto** *date*]
[**description** *"text"*]
{*date*|*day*|*calendar*|*request*|*"icalendar"*|*runcyclegroup*} [**,...**]
[**fdignore**|**fdnext**|**fdprev**]
[{(**at** *time* [+*n* **day**[**s**]]) |
(**schedtime** *time* [+*n* **day**[**s**]])}]
[**,...**]
[{**at** *time* [***timezone***|***tz*** *tzname*] [+*n* **day**[**s**]] |
schedtime *time* [***timezone***|***tz*** *tzname*] [+*n* **day**[**s**]]}]
[**until** *time* [**timezone**|**tz** *tzname*] [+*n* **day**[**s**]] [**onuntil** *action*]]
[**deadline** *time* [**timezone**|**tz** *tzname*] [+*n* **day**[**s**]]]
[**carryforward**]
[**matching** {**previous**|**sameday**|**relative from** [**+** | **-**] *time* **to** [**+** | **-**] *time*|
from *time* [+ | -*n* day[s]] **to** *time* [+ *n* day[s]] [**,...**]}]
[**follows** {[*netagent*::][*workstation*#]*jobstreamname*[**.***jobname* |**@**] [**previous**|
sameday|**relative from** [**+**|**-**] *time* **to** [**+**|**-**] *time*|
from *time* [+|-*n* day[s]] **to** *time* [+|-*n* day[s]]
]}] [,...]] [...]
[**keysched**]
[**limit** *joblimit*]
[**needs** { [*n*] [*workstation***#**]*resourcename* } [,...]] [...]
[**opens** { [*workstation*#]"*filename*" [(*qualifier*)] [,...] }] [...]
[**priority** *number* | **hi** | **go**]
[**prompt** {*promptname*|"[**:**|**!**]*text*"} [,...]] [...]
:

job-statement
    # comment
    [{**at** *time* [**timezone**|**tz** *tzname*] [**+***n* **day**[**s**]] |
    schedtime *time* [**timezone**|**tz** *tzname*] [**+***n* **day**[**s**]]}][**,...**]
    [**until** *time* [**timezone**|**tz** *tzname*] [**+***n* **day**[**s**]] [**onuntil** *action*]
    [**deadline** *time* [**timezone**|**tz** *tzname*] [**+***n* **day**[**s**]]] [**,...**]
    [**maxdur** *time* | *percentage* % **onmaxdur** *action*]
    [**mindur** *time* | *percentage* % **onmindur** *action*]
    [**every** *rate*]
    [**follows** {[*netagent*::][*workstation*#]*jobstreamname*{.*jobname* **@**} [**previous**|
      sameday|**relative from [+|-]** *time* **to [+|-]** *time* |
      from *time* [+|-*n* day[s]] **to** *time* [+|-*n* day[s]]
       ]}] [,...]] [...]
    [**confirmed**]
    [**critical**]
    [**keyjob**]
    [**needs** { [*n*] [*workstation***#**]*resourcename* } [,...]] [...]
    [**opens** { [*workstation*#]"*filename*" [**(***qualifier***)**] [**,**...] }] [...]
    [**priority** *number* | **hi** | **go**]
    [**prompt** {*promptname*|"[**:**|**!**]*text*"} [**,**...]] [...]

[*job-statement***...**]
end

Parâmetro

Sintaxe de definição do arquivo:

```
$parm
[tablename.]variablename "variablevalue"
```

Aviso

Sintaxe de definição do arquivo:

```
$prompt
promptname "[: | !]text"
```

Resource

Sintaxe de definição do arquivo:

```
$resource
workstation#resourcename units ["description" ]
```

Grupo do Ciclo de Execução

Sintaxe de definição do arquivo:

```
$runcyclegroup
runcyclegroupname ["description"]
      vartable tablename
        [freedays calendarname [-sa] [-su]]
  [on [runcycle name]
        [validfrom date] [validto date]
        [description
"text"]
        [vartable table_name]
       {date|day|calendar|request|"icalendar"} [,...]
```

[**fdignore** | **fdnext** | **fdprev**][**subset** *subsetname* **AND** | **OR**]
[({**at** *time* [+*n* **day**[**s**]] |
schedtime *time* [+*n* **day**[**s**]]}
[**until** *time* [+*n* **day**[**s**]] [**onuntil** *action*]]
[**deadline** *time* [+*n* **day**[**s**]])]]
[**,...**]
[**except** [**runcycle** *name*]
[**validfrom** *date*] [**validto** *date*]
[**description**
"*text*"]
{*date* | *day* | *calendar* | *request* | "*icalendar*"} [**,...**]
[**fdignore** | **fdnext** | **fdprev**][**subset** *subsetname* **AND** | **OR**]
[{(**at** *time* [+*n* **day**[**s**]])] |
(**schedtime** *time* [+*n* **day**[**s**]])}]
[**,...**]
[{**at** *time* [***timezone*** | ***tz*** *tzname*] [+*n* **day**[**s**]] |
schedtime *time* [***timezone*** | ***tz*** *tzname*] [+*n* **day**[**s**]]}]
[**until** *time* [**timezone** | **tz** *tzname*] [+*n* **day**[**s**]] [**onuntil** *action*]]
[**deadline** *time* [**timezone** | **tz** *tzname*] [+*n* **day**[**s**]]]
end

Tabela de Variáveis

Sintaxe de definição do arquivo:

vartable *tablename*
[**description** "*description*"]
[**isdefault**]
members
[*variablename* "*variablevalue*"]
...
[*variablename* "*variablevalue*"]
end

Para obter informações adicionais, consulte a Capítulo 6, "Customizando a Carga de Trabalho Utilizando Tabelas de Variáveis", na página 113.

Estação de Trabalho

Sintaxe de definição do arquivo:

cpuname *workstation* [**description**
"*description*"]
[**vartable** *table_name*]
os *os-type*
[**node** *hostname*] [**tcpaddr** *port*]
[**secureaddr** *port*] [**timezone** | **tz** *tzname*]
[**domain** *domainname*]
[**for maestro** [**host** *host-workstation* [**access** *method*]]
[**type fta** | **s-agent** | **x-agent** | **manager** | **broker** | **agent** |
pool | **d-pool** |
rem-engine]
[**ignore**]
[**autolink on** | **off**]
[**behindfirewall on** | **off**]
[**securitylevel enabled** | **on** | **force**]
[**fullstatus on** | **off**]
[**server** *serverid*]]

```
      [protocol http | htpps]
      [members [workstation] [...]]
 [requirements jsdl_definition]]
end

cpuname workstation [description description]
[vartable table_name]
os os-type
node hostname  [tcpaddr port]
[secureaddr port] [timezone|tz tzname]
[domain domainname]
[for maestro       [host host-workstation [access method]]
      [type fta | s-agent | x-agent | manager]
      [ignore]
      [autolink on | off]
      [behindfirewall on | off]
      [securitylevel enabled | on | force]
      [fullstatus on | off]
      [server serverid]]
end
```

Classe de Estação de Trabalho

Sintaxe de definição do arquivo:

```
cpuclass workstationclass [description "description"]
                                  [ignore]
      members [workstation | @] [...]
end
```

Definição do Usuário

Sintaxe de definição do arquivo:

```
username[workstation#][domain\]username
      password "password"end
```

Comandos do Composer

Esta seção descreve as operações que podem ser executadas no banco de dados utilizando o programa de interface de linha de comandos **composer** com a sintaxe:

```
composer [connection_parameters] [-defaultws twscpu]
       ["command[&[command]][...]"]
```

em que *connection_parameters*, se não forem fornecidos nos arquivos `localopts` ou `useropts`, são os seguintes:

```
[-file filename]|
[-host hostname]
[-port port_name]
[-protocol protocol_name]
[-proxy proxy_name]
[-proxyport proxy_port_number]
[-password user_password]
[-timeout seconds]
[-username user_name]
```

Consulte "Configurando Opções para Utilização das Interfaces com o Usuário" na página 53 para obter detalhes adicionais.

Essas operações podem ser executadas somente a partir da linha de comandos do cliente composer instalado.

Na Tabela 115 que exibe a lista de comandos que podem ser utilizados com o objeto de planejamento, *filename* indica um arquivo existente quando utilizado na sintaxe para os comandos **add** e **replace**, indica um arquivo não existente quando utilizado na sintaxe para o comando **create/extract**.

Tabela 115. Comandos do Composer

Comando	Sintaxe	Autorização do Usuário
add	{**add** | **a**} *filename* [**;unlock**]	*add* ou *modify*
authenticate	{**authenticate** | **au**} [**username=***username* **password=***password*]	
continuar	{**continue** | **c**}	
create extract	{**create** | **cr** | **extract** | **ext**} *filename* **from** {[**calendars** | **calendar** | **cal**=*calname*] | [**eventrule** | **erule** | **er**=*eventrulename*] | [**parms** | **parm** | **vb**=[*tablename*.]*variablename*] | [**vartable** | **vt**=*tablename*] | [**prompts** | **prom**=*promptname*] | [**resources** | **resource** | **res**=[*workstationname*#]*resourcename*] | [**runcyclegroup** | **rcg**=*runcyclegroupname*] | [**cpu**={*workstationname* | *workstationclassname* | *domainame*}] | [**workstation** | **ws**=*workstationname*] | [**workstationclass** | **wscl**=*workstationclassname*] | [**domain** | **dom**=*domainame*] | [**jobs** | **jobdefinition** | **jd**=[*workstationname*#]*jobname*] | [**sched** | **jobstream** | **js**= [*workstationname*#]*jstreamname* *[**valid from** date|**valid to** date |**valid in** date date]* *[**;full**]*] | [**users** | **user**=[*workstationame*#]*username* [;password]]} [**;lock**]	*display*
excluir	{**delete** | **de**} {[**calendars** | **calendar** | **cal**=*calname*] | [**domain** | **dom**=*domainame*] | [**eventrule** | **erule** | **er**=*eventrulename*] | [**parms** | **parm** | **vb**=[*tablename*.]*variablename*] | [**prompts** | **prom**=*promptname*] | [**resources** | **resource** | **res**=[*workstationname*#]*resourcename*] | [**runcyclegroup** | **rcg**=*runcyclegroupname*] | [**vartable** | **vt**=*tablename*] | [**wat**=*workloadapplicationtemplatename*] [**cpu**={*workstationname* [**;force**] | *workstationclassname* [**;force**]| *domainame*}] [**workstation** | **ws**=*workstationname*] [**;force**] | [**workstationclass** | **wscl**=*workstationclassname*] [**;force**] | [**jobs** | **jobdefinition** | **jd**=[*workstationname*#]*jobname*] | [**sched** | **jobstream** | **js**= [*workstationname*#]*jstreamname* *[**valid from** date|**valid to** date |**valid in** date date]*] | [**users** | **user**=[*workstationname*#]*username*]} [**;noask**]	*delete*
display	{**display** | **di**} {[**calendars** | **calendar** | **cal**=*calname*] | [**eventrule** | **erule** | **er**=*eventrulename*] | [**parms** | **parm** | **vb**=*variablename*.]*variablename*] | [**vartable** | **vt**=*tablename*] | [**prompts** | **prom**=*promptname*] | [**resources** | **resource** | **res**=[*workstationname*#]*resourcename*] | [**runcyclegroup** | **rcg**=*runcyclegroupname*] | [**cpu**={*workstationname* | *workstationclassname* | *domainame*}] [**workstation** | **ws**=*workstationname*] | [**workstationclass** | **wscl**=*workstationclassname*] | [**domain** | **dom**=*domainame*] | [**jobs** | **jobdefinition** | **jd**=[*workstationname*#]*jobname*] | [**sched** | **jobstream** | **js**= [*workstationname*#]*jstreamname* *[**valid from** date|**valid to** date |**valid in** date date]* *[**;full**]*] | [**users** | **user**=[*workstationname*#]*username*]} [**;offline**]	*display*
editar	{**edit** | **ed**} *filename*	
exit	{**exit** | **e**}	

Tabela 115. Comandos do Composer (continuação)

Comando	Sintaxe	Autorização do Usuário
list print	{**list** | **l**} {[**calendars** | **calendar** | **cal**=*calname*] | [**eventrule** | **erule** | **er**=*eventrulename*] | [**parms** | **parm** | **vb**=[*tablename.*]*variablename*] | [**vartable** | **vt**=*tablename*] | [**prompts** | **prom**=*promptname*] | [**resources** | **resource** | **res**=[*workstationname#*]*resourcename*] | [**runcyclegroup** | **rcg**=*runcyclegroupname*] | [**cpu**={*workstationname* | *workstationclassname* | *domainname*}] [**workstation** | **ws**=*workstationname*] | [**workstationclass** | **wscl**=*workstationclassname*] | [**domain** | **dom**=*domainname*] | [**jobs** | **jobdefinition** | **jd**=[*workstationname#*]*jobname*] | [**sched** | **jobstream** | **js**= [*workstationname#*]*jstreamname* *[**valid from** date* *__valid to__ date |__valid in__ date date]* | [**users** | **user**=[*workstationname#*]*username*]} [**;offline**]	*display*
lock	{**lock** | **lo**} {[**calendars** | **calendar** | **cal**=*calname*] | [**eventrule** | **erule** | **er**=*eventrulename*] [**parms** | **parm** | **vb**=[*tablename.*]*variablename*] | [**vartable** | **vt**=*tablename*] | [**prompts** | **prom**=*promptname*] | [**resources** | **resource** | **res**=[*workstationname#*]*resourcename*] | [**runcyclegroup** | **rcg**=*runcyclegroupname*] | [**cpu**={*workstationname* | *workstationclassname* | *domainname*}] [**workstation** | **ws**=*workstationname*] | [**workstationclass** | **wscl**=*workstationclassname*] | [**domain** | **dom**=*domainname*] | [**jobs** | **jobdefinition** | **jd**=[*workstationname#*]*jobname*] | [**sched** | **jobstream** | **js**= [*workstationname#*]*jstreamname* *[**valid from** date* | *__valid to__ date |__valid in__ date date]* | [**users** | **user**=[*workstationname#*]*username*]}	*modify*
modify	{**modify** | **m**} {[**calendars** | **calendar** | **cal**=*calname*] | [**eventrule** | **erule** | **er**=*eventrulename*] | [**parms** | **parm** | **vb**=[*tablename.*]*variablename*] | [**vartable** | **vt**=*tablename*] | [**prompts** | **prom**=*promptname*] | [**resources** | **resource** | **res**=[*workstationname#*]*resourcename*] | [**runcyclegroup** | **rcg**=*runcyclegroupname*] | [**cpu**={*workstationname* | *workstationclassname* | *domainname*}] [**workstation** | **ws**=*workstationname*] | [**workstationclass** | **wscl**=*workstationclassname*] | [**domain** | **dom**=*domainname*] | [**jobs** | **jobdefinition** | **jd**=[*workstationname#*]*jobname*] | [**sched** | **jobstream** | **js**= [*workstationname#*]*jstreamname* *[**valid from** date* | *__valid to__ date |__valid in__ date date]* *[**;full**]* | [**users** | **user**=[*workstationname#*]*username*]}	*modify* ou *add*
novo	**new** [**calendar** | **domain** | **eventrule** | **job** | **jobstream** | **parameter** | **prompt** | **resource** | **runcyclegroup** | **user** | **vartable** | **workstation** | **workstation_class**]	*add* ou *modify*
rename	{**rename** | **rn**} {**calendars** | **calendar** | **cal** | **parms** | **parm** | **vb** | **vartable** | **vt** | **prompts** | **prom** | **resorces** | **resource** | **res** | **runcyclegroup** | **rcg** | **workstation** | **ws** | **workstationclass** | **wscl** | **domain** | **dom** | **jobs** | **jobdefinition** | **jd** | **jobsched** | **jb** | **eventrule** | **erule** | **er** **sched** | **jobstream** | **js** | **users** | **user** } *old_object_identifier new_object_identifier*	*add* e *delete*
replace	{**replace** | **rep**} *filename* [**;unlock**]	*modify* ou *add*

Tabela 115. Comandos do Composer (continuação)

Comando	Sintaxe	Autorização do Usuário
destravar	{**unlock** | **u**} {[**calendars** | **calendar** | **cal**=*calname*] | [**eventrule** | **erule** | **er**=*eventrulename*] | [**parms** | **parm** | **vb**=[*tablename.*]*variablename*] | [**vartable** | **vt**=*tablename*] | [**prompts** | **prom**=*promptname*] | [**resources** | **resource** | **res**=[*workstationname#*]*resourcename*] | [**runcyclegroup** | **rcg**=*runcyclegroupname*] | [**cpu**={*workstationname* | *workstationclassname* | *domainname*}] [**workstation** | **ws**=*workstationname*] | [**workstationclass** | **wscl**=*workstationclassname*] | [**domain** | **dom**=*domainame*] | [**jobs** | **jobdefinition** | **jd**=[*workstationname#*]*jobname*] | [**sched** | **jobstream** | **js**= [*workstationname#*]*jstreamname* *[**valid from** date* | ***valid to** date* | ***valid in** date date]*] | [**users** | **user**=[*workstationname#*]*username*]} [**;forced**]	*modify* e *unlock*
validate	{**validate** | **val**} *filename* [**;syntax**]	

Gerenciando objetos no plano

Esta seção descreve as operações que podem ser executadas no plano utilizando o programa de interface de linha de comandos do **conman** com a sintaxe:

```
conman ["command[&[command]...] [&]"]
```

Comandos do Conman

Esta seção lista os comandos que podem ser executados a partir do programa **conman**.

Essa é a forma como você acessa a linha de comandos **conman**:

```
conman
[connection_parameters] ["command[&[command]...]
[&]"]
```

em que *connection_parameters*, se não forem fornecidos nos arquivos `localopts` ou `useropts`, são os seguintes:

```
[-file filename]
[-host hostname]
[-port port_name]
[-protocol protocol_name]
[-proxy proxy_name]
[-proxyport proxy_port_number]
[-password user_password]
[-timeout seconds]
[-username user_name]
```

Consulte “Configurando Opções para Utilização das Interfaces com o Usuário” na página 53 para obter detalhes adicionais.

Esse é a forma como você seleciona tarefas em comandos:

```
[workstation#]
{jobstreamname(hhmm[ date]) job|jobnumber}
[{+|~}jobqualifier[...]]
```

ou:

```
[workstation#]
jobstream_id.
job
[{+|~]jobqualifier[...]]
;schedid
```

Essa é a forma como você seleciona fluxos de tarefas em comandos:

```
[workstation#]
jobstreamname(hhmm[ date])
[{+|~}jobstreamqualifier[...]]
```

ou:

```
[workstation#]
jobstream_id
 ;schedid
```

É possível executar esses comandos a partir de diferentes tipos de estações de trabalho. Nesta tabela:

D significa gerenciadores de domínios e agentes tolerantes a falhas.

S significa agentes padrão.

Para cada comando você localiza o nome, a sintaxe, o tipo de estação de trabalho a partir de onde é possível emitir o comando e a autorização necessária, se houver.

Tabela 116. Comandos que Podem Ser Executados a partir de Conman

Comando	Sintaxe	Tipos de estação de trabalho	Autorização do Usuário
adddep job	{**adddep job** | **adj**} *jobselect* **;***dependency*[**;**...] [**;noask**]	D	*adddep* - (*use* ao utilizar prompts e needs)
adddep sched	{**adddep sched** | **ads**} *jstreamselect* **;***dependency*[**;**...] [**;noask**]	D	*adddep* - (*use* ao utilizar prompts e needs)
altpass	**altpass** [*workstation***#**] *username* [**;"***password***"**]	D	*altpass*
altpri	{**altpri** | **ap**} *jobselect* | *jstreamselect* [**;***pri*] [**;noask**]	D	*altpri*
bulk_discovery	{**bulk_discovery** | **bulk**}	D	*display*
cancel job	{**cancel job** | **cj**} *jobselect* [**;pend**] [**;noask**]	D	*cancel*
cancel sched	{**cancel sched** | **cs**} *jstreamselect* [**;pend**] [**;noask**]	D	*cancel*
checkhealthstatus	{**checkhealthstatus** | **chs**} [*workstation*]	M,F,S	
confirmar	{**confirm** | **conf**} *jobselect* **;**{**succ** | **abend**} [**;noask**]	D	*confirmar*
console	{**console** | **cons**} [**sess** | **sys**] [**;level=***msglevel*]	F-S	*console*
continuar	{**continue** | **cont**}	F-S	
deldep job	{**deldep job** | **ddj**} *jobselect* **;***dependency*[**;**...] [**;noask**]	D	*deldep*
deldep sched	{**deldep sched** | **dds**} *jstreamselect* **;***dependency*[**;**...] [**;noask**]	D	*deldep*
deployconf	{**deployconf** | **deploy**} [*domain***!**]*workstation*	F,S	Permissão para ações *start* nos objetos *cpu*
display	{**display file** | **df**} *filename* [**;offline**] {**display job** | **dj**} *jobselect* [**;offline**] {**display sched** | **ds**} *jstreamselect* [**valid** {**at** *date* | **in** *date date*} [**;offline**]	F-S[1]	*display*
exit	{**exit** | **e**}	F-S	
limite	{**fence** | **f**} *workstation* **;***pri* [**;noask**]	D	*fence*
help (somente para UNIX)	{**help** | **h**} {*command*|*keyword*}	F-S	
kill	{**kill** | **k**} *jobselect* [**;noask**]	D	*kill*
limit cpu	{**limit cpu** | **lc** } *workstation* **;***limit* [**;noask**]	D	*limite*
limit sched	{**limit sched** | **ls** } *jstreamselect* **;***limit* [**;noask**]	D	*limite*
link	{**link** | **lk**} [*domain***!**]*workstation* [**;noask**]	F-S	*link*

Tabela 116. Comandos que Podem Ser Executados a partir de Conman (continuação)

Comando	Sintaxe	Tipos de estação de trabalho	Autorização do Usuário
listsym	{**listsym** | **lis**} [**trial** | **forecast**] [**;offline**]	D	
recall	{**recall** | **rc**} [*workstation*] [**;offline**]	D	*display*
redo	{**redo** | **red**}	F-S	
release job	{**release job** | **rj**} *jobselect* [**;***dependency*[**;**...]] [**;noask**]	D	*release*
release sched	{**release sched** | **rs**} *jstreamselect* [**;***dependency*[**;**...]] [**;noask**]	D	*release*
reply	{**reply** | **rep**} { *promptname* | [*workstation***#**]*msgnum*} **;***responder* [**;noask**]	D	*reply*
executar novamente	{**rerun** | **rr**} *jobselect* [**;from=**[*wkstat***#**]*job* [**;at=***time*] [**;pri=***pri*]] [**;step=***step*] [**;noask**]	D	*rerun*
recurso	{**resource** | **reso**} [*workstation***#**] *resource***;***num* [**;noask**]	D	*resource*
setsym	{**setsym** | **set**} [**trial** | **forecast**] [*filenum*]	D	
showcpus	{**showcpus** | **sc**} [[*domain***!**]*workstation*] [**;info** | **;link**] [**;offline**]	F-S	*list*[2]
showdomain	{**showdomain** | **showd**} [*domain*] [**;info**] [**;offline**]	F-S	*list*[2]
showfiles	{**showfiles** | **sf**} [[*workstation***#**]*file*] [**;***state*[**;**...]] [**;keys**] [**;offline**] {**showfiles** | **sf**} [[*workstation***#**]*file*] [**;***state*[**;**...]] [**;deps**[**;keys** | **info** | **logon**]] [**;offline**]	D	
showjobs	{**showjobs** | **sj**} [*jobselect*] [**;deps**[**;keys** | **info** | **logon**]] [**;short** | **single**] [**;offline**] [**;showid**] [**;props**] {**showjobs** | **sj**} [*jobselect* | [**workstation#**]*jobnumber.hhmm*] [**;stdlist**[**;keys**]] [**;short** | **single**] [**;offline**] [**;showid**] [**;props**]	D	*list*[2]
showprompts	{**showprompts** | **sp**} [*promptselect*] [**;keys**] [**;offline**] {**showprompts** | **sp**} [*promptselect*] [**;deps**[**;keys** | **info** | **logon**]][**;offline**]	D	*list*[2]
showresources	{**showresources** | **sr**} [[*workstation***#**]*resourcename*] [**;keys**] [**;offline**] {**showresources** | **sr**} [[*workstation***#**]*resourcename*] [**;deps**[**;keys** | **info** | **logon**]] [**;offline**]	D	*list*[2]
showschedules	{**showscheds** | **ss**} [*jstreamselect*] [**;keys**] [**;offline**] [**;showid**] {**showscheds** | **ss**} [*jstreamselect*] [**;deps**[**;keys** | **info** | **logon**]] [**;offline**] [**;showid**]	D	*list*[2]
shutdown	{**shutdown** | **shut**} [**;wait**]	F-S	*shutdown*
início	**start** [*domain***!**]*workstation* [**;mgr**] [**;noask**] [**;demgr**]	F-S	*start*

Tabela 116. Comandos que Podem Ser Executados a partir de Conman (continuação)

Comando	Sintaxe	Tipos de estação de trabalho	Autorização do Usuário
startappserver	**startappserver**[*domain***!**]*workstation* [**;wait**]	F-S	Permissão para ações *start* nos objetos *cpu*
startevtp	{**starteventprocessor** | **startevtp**} [*domain***!**]*workstation*	M[4]	Permissão para ações *start* nos objetos *cpu*
startmon	{**startmon** | **startm**} [*domain***!**]*workstation* [**;noask**]	F-S	Permissão para ações *start* nos objetos *cpu*
status	{**status** | **stat**}	F-S	appserver
parar	**stop** [*domain***!**]*workstation* [**;wait**] [**;noask**]	F-S	*parar*
stop ;progressive	**stop ;progressive**		*parar*
stopappserver	{**stopappserver** | **stopapps**} [*domain***!**]*workstation* [**;wait**]	F-S	Permissão para ações *stop* nos objetos *cpu*
stopevtp	{**stopeventprocessor** | **stopevtp**} [*domain***!**][*workstation*]	M[4]	Permissão para ações *stop* nos objetos *cpu*
stopmon	{**stopmon** | **stopm**} [*domain***!**]*workstation* [**;wait**] [**;noask**]	F-S	Permissão para ações *stop* nos objetos *cpu*
submit docommand	{**submit docommand** | **sbd**} [*workstation***#**]**"***cmd***"** [**;alias**[**=***name*]] [**;into=**[*workstation***#**] {*jobstream_id***;schedid** | *jobstreamname* ([*hhmm*[*date*]])}] [*joboption*[**;**...]]	F-S	*submit* - (*use* ao utilizar prompts e needs)
submit file	{**submit file** | **sbf**} **"***filename***"** [**;alias**[**=***name*]] [**;into=**[*workstation***#**](*jobstream_id* **;schedid** | *jobstreamname*([*hhmm*[*date*]])}] [*joboption*[**;**...]] [**;noask**]	F-S	*submit* - (*use* ao utilizar prompts e needs)
submit job	{**submit job** | **sbj**} [*workstation***#**]*jobname* [**;alias**[**=***name*]] [**;into=**[*workstation***#**](*jobstream_id* **;schedid** | *jobstreamname*([*hhmm*[*date*]])}] [*joboption*[**;**...]] [**;vartable=***tablename*] [**;noask**]	F-S[3]	*submit* - (*use* ao utilizar prompts e needs)
submit sched	{**submit sched** | **sbs**} [*workstation***#**]*jstreamname* [**;alias**[**=***name*]] [*jstreamoption*[**;**...]] [**;vartable=***tablename*] [**;noask**]	F-S[3]	*submit* - (*use* ao utilizar prompts e needs)
switchevtp	{**switcheventprocessor** | **switchevtp**} *workstation*	M[4]	Permissão para ações *start* e *stop* nos objetos *cpu*
switchmgr	{**switchmgr** | **switchm**} *domain***;***newmgr*	D	*start stop*
system	[**:** | **!**] *system-command*	F-S	
tellop	{**tellop** | **to**} [*text*]	F-S	
unlink	**unlink** [*domain***!**]*workstation* [**;noask**]	F-S	*unlink*
version	{**version** | **v**}	F-S	

em que:

(1) Indica que só é possível exibir arquivos em um agente padrão.

(2) Você deve ter acesso **list** para o objeto que está sendo mostrado se a opção ***enListSecChk*** tiver sido configurada para **yes** no gerenciador de domínios principal quando o plano de produção tiver sido criado ou estendido.

(3) Indica que é possível utilizar subimt job (**sbj**) e submit sched (**sbs**) em um agente padrão, utilizando os parâmetros de conexão ou especificando as configurações no arquivo `useropts` ao chamar a linha de comandos **conman**.

(4) É possível utilizar este comando em gerenciadores de domínio principais e principais de backup bem como em estações de trabalho instaladas como principais de backup mas utilizadas como agente tolerante a falhass comuns.

Comandos utilitários

Esta seção contém a lista dos comandos utilitários que podem ser executados a partir do prompt de comandos do sistema operacional. Os comandos utilitários são divididos em três grupos, aqueles que podem ser executados nos sistemas operacionais UNIX e Windows, aqueles que podem ser executados somente no UNIX e aqueles que podem ser executados no Windows.

Comandos utilitários disponíveis para os sistemas operacionais UNIX e Windows

Tabela 117. Comandos Utilitários Disponíveis para o UNIX e oWindows

Comando	Sintaxe
às	**at -V** | **-U** **at -s** *jstream* | **-q** *queue time-spec*
batch	**batch -V** | **-U** **batch** [**-s** *jstream*]
cpuinfo	**cpuinfo -V** | **-U** **cpuinfo** *workstation* [*infotype*] [...]
datecalc	**datecalc -V** | **-U** **datecalc** *base-date* [*offset*] [**pic** *format*][**freedays** *Calendar_Name* [**-sa**] [**-su**]] **datecalc -t** *time* [*base-date*] [*offset*] [**pic** *format*] **datecalc** *yyyymmddhhtt* [*offset*] [**pic** *format*]
excluir	**delete -V** | **-U** **delete** *filename*
evtdef	**evtdef -U** | **-V** **evtdef** [*connection parameters*] **dumpdef** *file-path* **evtdef** [*connection parameters*] **loaddef** *file-path*
evtsize	**evtsize -V** | **-U** **evtsize** *filename size* **evtsize -compact** *filename* [*size*] **evtsize -info** *filename* **evtsize -show** *filename* **evtsize -info** | **-show pobox**
jobinfo	**jobinfo -V** | **-U** **jobinfo** *job-option* [...]
jobstdl	**jobstdl -V** | **-U** **jobstdl** [**-day** *num*] [{**-first** | **-last** | **-num** *n* | **-all**}] [**-twslog**] [{**-name** ["*jobstreamname* [(*hhmm* *date*),(*jobstream_id*)].]*jobname*" | *jobnum* | **-schedid** *jobstream_id.jobname*}]
maestro	**maestro** [**-V** | **-U**]

Tabela 117. Comandos Utilitários Disponíveis para o UNIX e oWindows (continuação)

Comando	Sintaxe
makecal	**makecal** [**-c** *name*] **-d** *n* | **-e** | {**-f 1** | **2** | **3 -s** *date*} | **-l** | **-m** | **-p** *n* | {**-r** *n* **-s** *date*} | **-w** *n* [**-i** *n*] [**-x** | **-z**][**-freedays** *Calendar_Name* [**-sa**] [**-su**]]
morestdl	**morestdl -V** | **-U** **morestdl** [**-day** *num*] [**-first** | **-last** | **-num** *n* | **-all**] [**-twslog**] [{**-name** ["*jobstreamname* [(*hhmm date*),(*jobstream_id*)].]*jobname*" | *jobnum* | **-schedid** *jobstream_id.jobname*}]
param	**param -u** | **-V** **param** {**-c** | **-ec**} [*file.section.* | *file.* | *section.*] *variable* [*value*] **param** [*file.section.* | *file.* | *section.*] *variable* **param** {**-d** | **-fd**} [*file.section.* | *file.* | *section.*] *variable*
parms	**parms** {[**-V** | **-U**] | **-build**} **parms** {**-replace** | **-extract**} *filename* **parms** [**-d**]*parametername***parms -c** *parametername value*
release	**release -V** | **-U** **release** [**-s**] [*workstation***#**]*resourcename* [*count*]
rmstdlist	**rmstdlist -V** | **-U** **rmstdlist** [**-p**] [*age*]
sendevent	**sendevent -V** | **?** | **-help** | **-U** | **-usage** **sendevent** [**-hostname** *hostname*] [{**-port** | **-sslport**} *port*] *eventType* *source* [[*attribute=value*]...] Em ambientes dinâmicos: **sendevent** [**-hostname** *hostname*] [**-port** *port*] *eventType* *source* [[*attribute=value*]...]
showexec	**showexec** [**-V** | **-U** | **INFO**]
ShutDownLwa	**ShutDownLwa**
StartUp	**StartUp** [**-V** | **-U**]
StartUpLwa	**StartUpLwa**
tws_inst_pull_info	**tws_inst_pull_info -twsuser** *userid* **-log_dir_base** *path* [**-u** | [**-run_db2_module** [y|n] | **-extract_db_defs** [y|n] | **-date** *yyyymmdd*]

Comandos utilitários disponíveis somente para o sistema operacional UNIX

Tabela 118. Comandos Utilitários Disponíveis somente para o UNIX

Comando	Sintaxe
às	**at -V** | **-U** **at -s***jstream* | **-q***queuetime-spec*

Tabela 118. Comandos Utilitários Disponíveis somente para o UNIX (continuação)

Comando	Sintaxe
batch	**batch -V | -U** **batch** [**-s** *jstream*]
showexec	**showexec** [**-V | -U | -info**]
version	**version -V | -U | -h** **version** [**-a**] [**-f** *vfile*] [*file* [...]]

Comandos utilitários disponíveis somente para o sistema operacional Windows

Tabela 119. Comandos Utilitários Disponíveis somente para o Windows

Comando	Sintaxe
listproc **(NÃO SUPORTADO)**	**listproc**
killproc **(NÃO SUPORTADO)**	**killproc** *pid*
shutdown	**shutdown** [**-V | -U**] [**-appsrv**]

Comandos de relatório

Esta seção contém uma lista e a sintaxe de comandos de relatório e de programas de extração de relatórios. Esses comandos são executados a partir do prompt de comandos do sistema operacional.

Comandos de relatório

Tabela 120. Comandos de relatório

Per.	saída produzida	Sintaxe
rep1	Relatório 01 - Listagem de Detalhes do Job	**rep**[*x*] [**-V|-U**]
rep2	Relatório 02 - Listagem do Prompt	**rep**[*x*] [**-V|-U**]
rep3	Relatório 03 - Listagem de Calendário	**rep**[*x*] [**-V|-U**]
rep4a	Relatório 04A - Listagem de Parâmetro	**rep**[*x*] [**-V|-U**]
rep4b	Relatório 04B - Listagem de Recurso	**rep**[*x*] [**-V|-U**]
rep7	Relatório 07 - Listagem do Histórico do Job	**rep7 -V|-U** **rep7** [**-c** *wkstat*] [**-s** *jstream_name*] [**-j** *job*] [**-f** *date* **-t** *date*] [**-l**]
rep8	Relatório 08 - Histograma do Job	**rep8 -V|-U** **rep8** [**-f** *date* **-b** *time* **-t** *date* **-e** *time*] [**-i** *file*] [**-p**] **rep8** [**-b** *time* **-e** *time*] [**-i** *file*] [**-p**]
rep11	Relatório 11 - Agendamento de Produção Planejada	**rep11 -V|-U** **rep11** [**-m***mm*[*yy*] [...]] [**-c** *wkstat* [...]] [**-s** *jstream_name*] [**-o** *output*]

Tabela 120. Comandos de relatório (continuação)

Per.	saída produzida	Sintaxe
reptr	Relatório 09A - Resumo de Produção Planejada Relatório 09B - Detalhe de Produção Planejada Relatório 10A - Resumo de Produção Real Relatório 10B - Detalhe de Produção Real	**reptr** [**-V**|**-U**] **reptr -pre** [**-**{**summary**| **detail**}] [*symfile*] **reptr -post** [**-**{**summary** | **detail**}] [*logfile*]
xref	Relatório 12 - Relatório de Referência Cruzada	**xref** [**-V**|**-U**] **xref** [**-cpu** *wkstat*] [**-s** *jstream_name*] [**-depends**|**-files**|**-jobs**|**-prompts**|**-resource**|**-schedules**|**-when** [...]]

Programas de Extração de Relatório

Tabela 121. Programas de Extração de Relatório

Programa de Extração	Utilizado para gerar	Sintaxe
jbxtract	Relatório 01 Relatório 07	**jbxtract** [**-V** | **-U**] [**-j** *job*] [**-c** *wkstat*] [**-o** *output*]
prxtract	Relatório 02	**prxtract** [**-V** | **-U**] [**-o** *output*] **prxtract**[**-V** | **-U**] [**-m** *mm*[*yyyy*]] [**-c** *wkstat*] [**-o** *output*]
caxtract	Relatório 03	**caxtract** [**-V** | **-U**] [**-o** *output*]
paxtract	Relatório 04A	**paxtract** [**-V** | **-U**] [**-o** *output*]
rextract	Relatório 04B	**rextract** [**-V** | **-U**] [**-o** *output*]
r11xtr	Relatório 11	**r11xtr** [**-V** | **-U**] [**-m** *mm*[*yyyy*]] [**-c** *wkstat*] [**-o** *output*] [**-s** *jstream_name*]
xrxtrct	Relatório 12	**xrxtrct** [**-V** | **-U**]

Apêndice D. Definindo e Gerenciando Tarefas de Ramificação Genéricas

O Tivoli Workload Scheduler fornece uma ampla faixa de funções de planejamento. É possível estender estas funções para necessidades específicas, usando uma solução customizada adicional, chamada tarefa de ramificação genérica.

Para obter informações sobre como usar essa solução, consulte as seções a seguir:

- "Introdução"
- "Cenários de Amostra" na página 733
- "Trabalhando com a Tarefa de Ramificação" na página 766
- "Especificando os Parâmetros da Tarefa de Ramificação" na página 770
- "Notas Importantes sobre a Tarefa de Ramificação" na página 780

Nota: A tarefa de ramificação genérica não é uma parte nativa do Tivoli Workload Scheduler, mas é suportada pelo Suporte de Software IBM. Ele é fornecido apenas com as mensagens em inglês e regras específicas de convenção de nomenclatura (para obter informações detalhadas, consulte "Colocando a Tarefa de Ramificação no Fluxo de Tarefas" na página 768.

É altamente recomendável usar a tarefa de ramificação primeiro em seu ambiente de teste. A mesma recomendação é válida também para os cenários de amostra. Apenas depois de produzir uma execução sem erros da tarefa de ramificação ou do cenário de amostra, você deverá mudar para o ambiente de produção.

Introdução

Definir uma tarefa de ramificação genérica para avaliar um status de tarefa específica ou saída e, com base nas condições especificadas, decidir quais tarefas executar no fluxo de tarefas.

O Tivoli Workload Scheduler oferece recursos de relatório, planejamento orientado a eventos e muitos outros recursos. No entanto, algumas tarefas podem ser executadas apenas com programação adicional, como a avaliação lógica de um fluxo do processo em um fluxo de tarefas.

Use uma tarefa de ramificação genérica para avaliar um status de tarefa específica ou saída e decida quais tarefas executar em um fluxo de tarefas. As condições que determinam quais tarefas serão executadas podem ser muito simples (por exemplo, o predecessor foi encerrado com o estado SUCC ou ABEND) ou resultarem de uma operação booleana ou aritmética complexa.

A Figura 39 na página 728 mostra os conceitos básicos de uma tarefa de ramificação genérica.

© Copyright IBM Corp. 1999, 2014

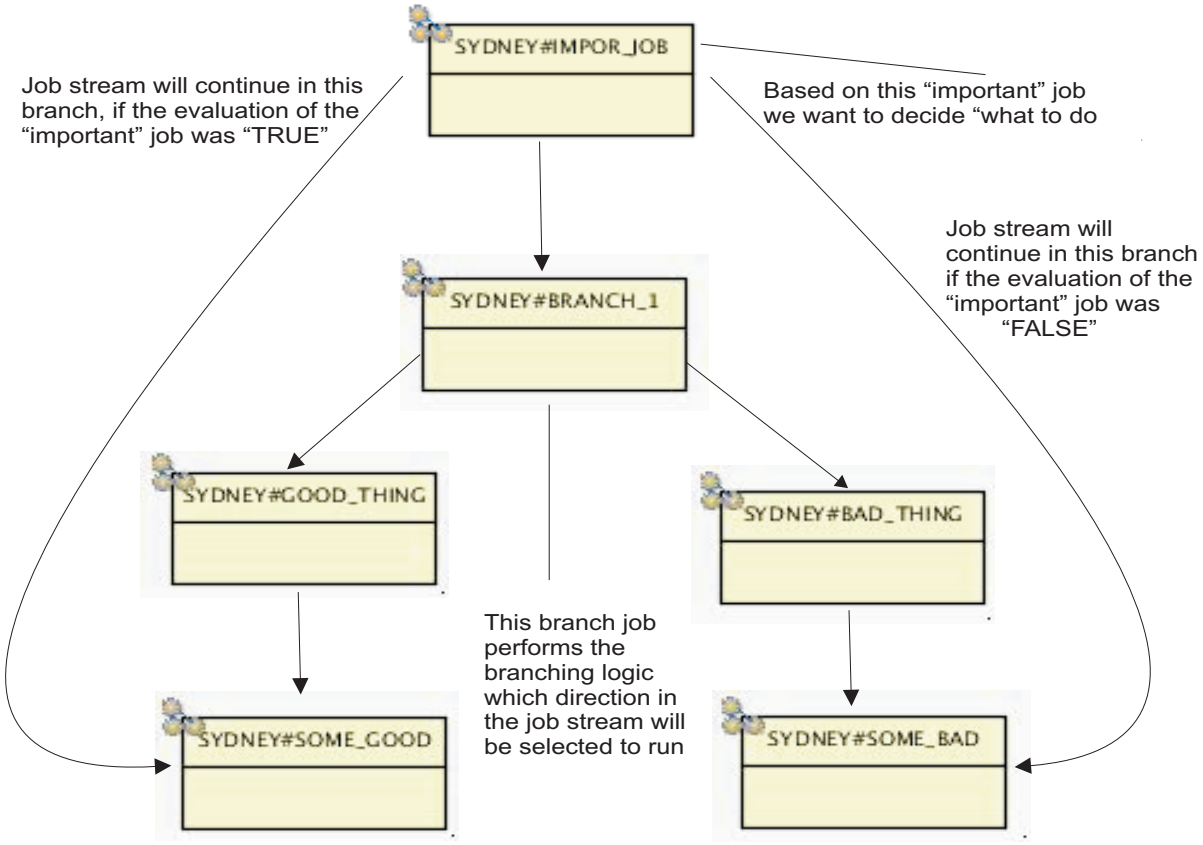


Figura 39. Propósito da tarefa de ramificação

A função de tarefa de ramificação genérica não está incluída no Tivoli Workload Scheduler, mas foi desenvolvida por especialistas da IBM durante várias implementações do Tivoli Workload Scheduler do cliente. A tarefa de ramificação genérica usa a interface aberta do Tivoli Workload Scheduler.

É possível implementar a ramificação simples, com base no status da tarefa predecessora (SUCC ou ABEND), em um período de tempo muito curto. No entanto, para cenários mais complexos, por exemplo, ao procurar um padrão e fazer comparações numéricas, você deve especificar alguns parâmetros de entrada.

Terminologia

Os seguintes termos são usados para descrever tarefas de ramificação:

Pai (às vezes também referido como tarefa avaliada)
A tarefa cujo status ou outras propriedades você deseja avaliar.

Condição
A condição que está sendo executada em relação à tarefa-pai. A condição pode ser tão simples quanto*consultar o status do pai* ou mais complexa.

Filho válido
O sucessor mais próximo da tarefa de ramificação que será executado se a condição for avaliada como TRUE.

Filho inválido
O sucessor mais próximo da tarefa de ramificação que será interrompido se a condição for avaliada como FALSE.

Ramificação
Uma sequência de tarefas que são ordenadas usando a dependência FOLLOWS.

Executar ramificação
A ramificação selecionada para execução, com base no resultado da condição. A ramificação de execução começará com o filho válido se a condição for avaliada como TRUE ou com o filho inválido se a condição for avaliada como FALSE.

Parar ramificação
A ramificação selecionada para parar, com base no resultado da condição. A ramificação de parada começará com o filho inválido se a condição for avaliada como TRUE ou com o filho válido se a condição for avaliada como FALSE

Parâmetro de entrada
Argumento que é transmitido para a tarefa de ramificação específica. Para alguns parâmetros, se não for especificado nenhum valor, o padrão será usado.

Sufixo da tarefa de ramificação
Diferencia diversas ocorrências de tarefas de ramificação no mesmo fluxo de tarefas.

A Figura 40 na página 730 ilustra os seguintes termos, que são usados para definir um fluxo de tarefas gerenciado por uma ou mais tarefas de ramificação:

- Pai
- Filho válido e ramificação válida
- Filho inválido e ramificação inválida
- Onde a condição reside

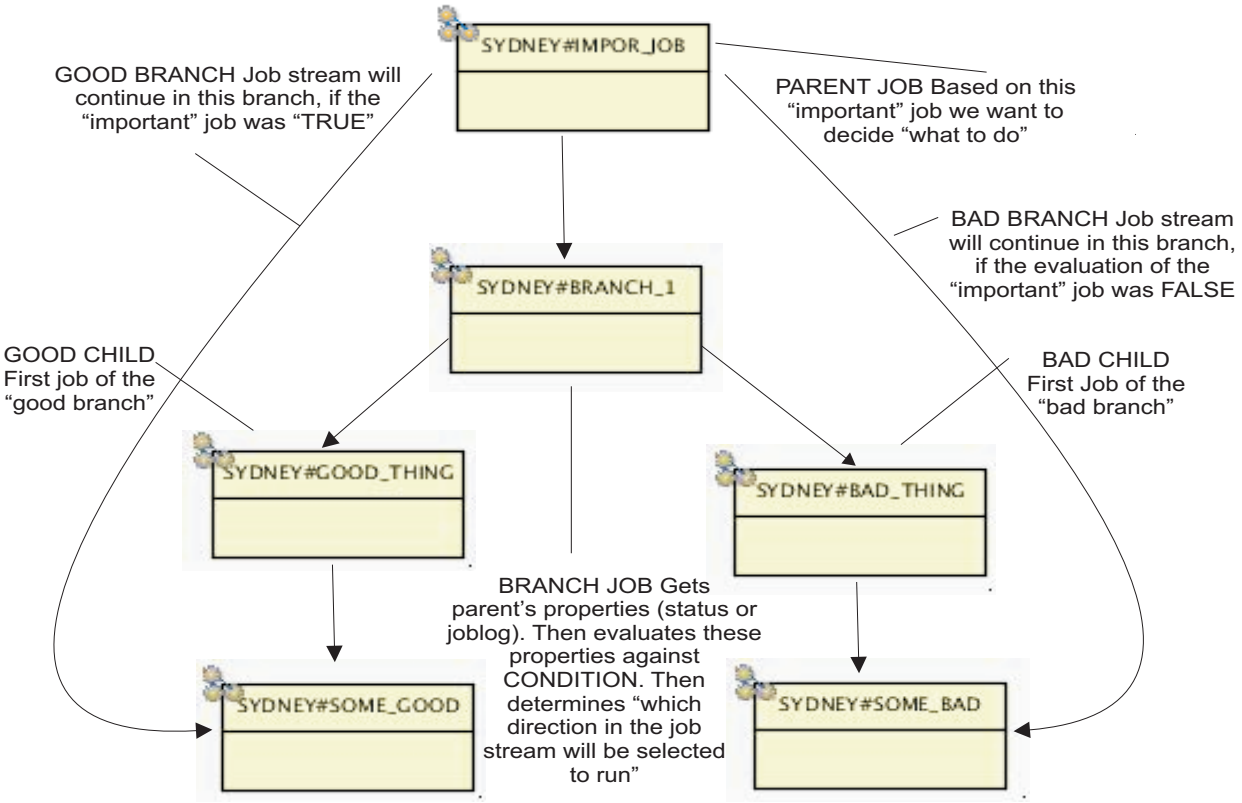


Figura 40. Termos relacionados à definição de fluxo de tarefas

A Figura 41 na página 731 ilustra termos adicionais que são usados durante a execução do fluxo de tarefas:

- Executar ramificação
- Parar ramificação

Ela também mostra a diferença entre os termos válido e inválido, e execução e parada. Esses termos são os mesmos apenas quando CONDITION=TRUE. Se CONDITION=FALSE, a ramificação de execução corresponderá à ramificação inválida e a ramificação de parada corresponderá à ramificação válida. Este é o conceito da lógica de avaliação de tarefa de ramificação.

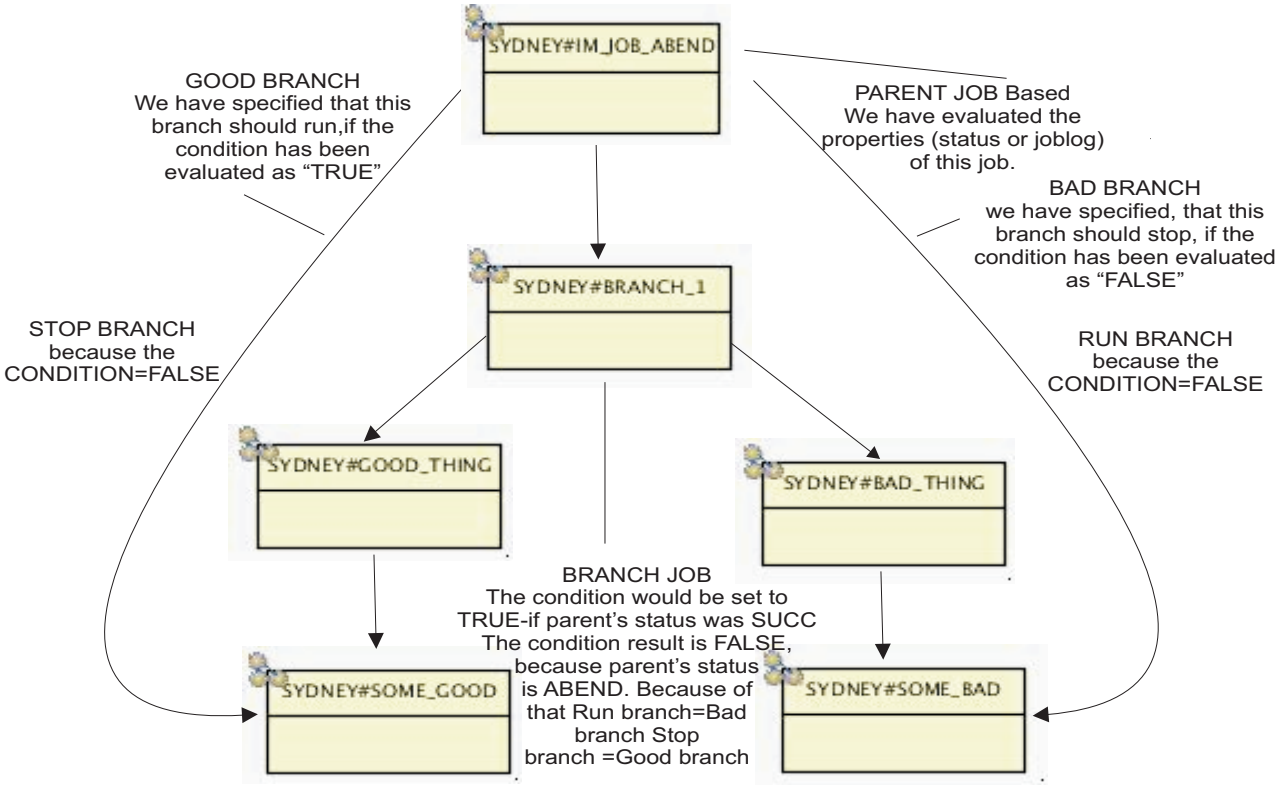


Figura 41. Termos relacionados à execução do fluxo de tarefas (instância do fluxo de tarefas concreta)

Recursos da Tarefa de Ramificação

Para descrever os recursos da tarefa de ramificação genérica, o processo de ramificação pode ser dividido em dois processos:

- **Avaliação**: A tarefa de ramificação sempre tem seu pai como a entrada. Em seguida, com base nos parâmetros especificados, ela cria uma das seguintes subcondições:
  - Verifica se o pai foi encerrado com o estado SUCC ou ABEND.
  - Cria uma condição complexa a partir de uma ou mais das seguintes subcondições:
    - Obtém o log da tarefa-pai e procura uma linha com um padrão de texto especificado (que é transmitido para a tarefa de ramificação como parâmetro de entrada).
    - (opcional) Procura outro padrão na mesma linha.
    - (opcional) Procura um valor numérico na mesma linha. Este valor numérico é comparado com um número especificado usando um operador aritmético especificado (o número e o operador aritmético são transmitidos para a tarefa de ramificação como parâmetros de entrada).

    Estas subcondições são associadas usando os operadores booleanos AND ou OR. Em seguida, a condição complexa é avaliada imediatamente.

  O resultado da avaliação (mesmo que a condição seja simples ou complexa) é TRUE ou FALSE.

- **Ação**: Com base no resultado da condição, a tarefa de ramificação decide qual ramificação executar e qual parar. Em seguida, ela executa ações diferentes na ramificação de execução e na ramificação de parada, da seguinte forma.
  - As possíveis ações na ramificação de execução são:

    DO NOTHING
    A ramificação de execução é executada.

    RELEASE
    Se a primeira tarefa da ramificação de execução estiver retida, ela será liberada.
  - As possíveis ações na ramificação de parada são:

    CANCEL
    Todas as tarefas na ramificação de parada são canceladas.

    PAUSE
    A ramificação de parada é suspensa. As tarefas na ramificação de parada não podem ser executadas porque seu predecessor está retido.
  - Além disso, há uma ação especial chamada SIGNAL. Esta ação apenas grava uma recomendação para a confirmação da tarefa de ramificação no log da tarefa. Você cria uma tarefa de ramificação SIGNAL configurando o sinalizador `Requires confirmation`.

    Para obter informações detalhadas sobre a tarefa de ramificação SIGNAL, consulte Cenário de Ação de Sinal

É possível combinar critérios de avaliação e ações consecutivas de qualquer maneira.

Vantagens da Tarefa de Ramificação

As principais vantagens do uso da tarefa de ramificação genérica são:

- Apenas uma tarefa de ramificação está definida no banco de dados do Tivoli Workload Scheduler.

  A tarefa de ramificação genérica é representada por uma definição de tarefa que aponta para um shell script. O script não usa nenhum argumento de linha de comandos que aponta para o pai e filhos da tarefa de ramificação. As informações sobre o pai (predecessor) e os filhos (sucessores) são avaliadas automaticamente.
- Não será necessário especificar nenhum parâmetro de entrada se você usar os cenários de tarefa de ramificação mais comuns.

  Não é necessário especificar parâmetros ao avaliar o estado do resultado da tarefa-pai (SUCC ou ABEND). Você coloca a tarefa de ramificação no fluxo de tarefas, vincula as dependências FOLLOWS e designa às tarefas-filhas os nomes específicos que identificam o filho válido e o filho inválido.
- A tarefa de ramificação retorna um log da tarefa estruturado que contém informações detalhadas sobre o ambiente da tarefa de ramificação, parâmetros de entrada, condição avaliada, ações executadas. Ao ler o log da tarefa, é possível verificar facilmente quais atividades foram executadas pela tarefa de ramificação.
- A tarefa de ramificação usa a representação de objetos do Tivoli Workload Scheduler no Plano Atual. Isso significa que todos os objetos no Plano (por exemplo, tarefas e fluxos de tarefas) são referenciados usando a palavra-chave `schedid`. Você assegura que todas as ações ativadas nos objetos do Plano apontem para o objeto exclusivo.

Esta funcionalidade é alavancada em todas as ocorrências do fluxo de tarefas no Plano Atual:

- Qualquer ocorrência do fluxo de tarefas enviada sem especificar o alias.
- Qualquer ocorrência do fluxo de tarefas enviado especificando o alias.
- A tarefa de ramificação genérica é executada em gerenciadores de domínio principal do UNIX e Windows. Como a tarefa de ramificação é gravada no shell script (portanto, é executada nativamente no UNIX) em sistemas operacionais Windows, você deve usar um interpretador de shell UNIX.
- A tarefa de ramificação genérica é executada mesmo que o fluxo de tarefas esteja definido na classe de estação de trabalho. A própria tarefa de ramificação deve ser definida no gerenciador de domínio principal.

Cenários de Amostra

Os cenários de amostra descrevem várias tarefas de ramificação, introduzindo os principais conceitos e explicando o uso da tarefa de ramificação.

Cada cenário é descrito de acordo com a seguinte estrutura:

- Uso de cenário.
- Definição de fluxo de tarefas de amostra.
- Log da tarefa de ramificação genérica.
- Parâmetros necessários para executar a ramificação.
- Como colocar o fluxo de tarefas no fluxo de tarefas e renomear nomes de tarefas-filhas, se necessário.

Uma tarefa de ramificação genérica baseada no tipo de condição se difere de uma tarefa de ramificação genérica baseada no tipo de ação, conforme a seguir:

Condição
Especifica os critérios para determinar a ramificação de execução (as tarefas que devem continuar) e a ramificação de parada (as tarefas que devem parar).

Ação Especifica o que fazer na ramificação de execução e na ramificação de parada. É possível executar dois tipos de ações na ramificação de parada e dois tipos de ações na ramificação de execução. Além disso, há uma ação de *sinalização* especial.

Cenários Baseados em Tipo de Condição

Use uma tarefa de ramificação genérica baseada no tipo de condição para especificar os critérios que determinam a ramificação de execução e a ramificação de parada.

Para cenários baseados no tipo de ação, consulte Cenários Baseados no Tipo de Ação.

Cenário de Ramificação Simples

Use a ramificação simples para avaliar o status de uma tarefa.

Uso de Ramificação Simples

Com base no status retornado pela tarefa que está sendo avaliada, é executada uma ramificação específica: se a tarefa for encerrada no status SUCC, a ramificação

válida será executada; se a tarefa for ENCERRADA DE FORMA ANORMAL, a ramificação inválida será executada.

Encerramento de Ramificação Simples no Estado SUCC

A Figura 42 mostra a definição de fluxo de tarefas.

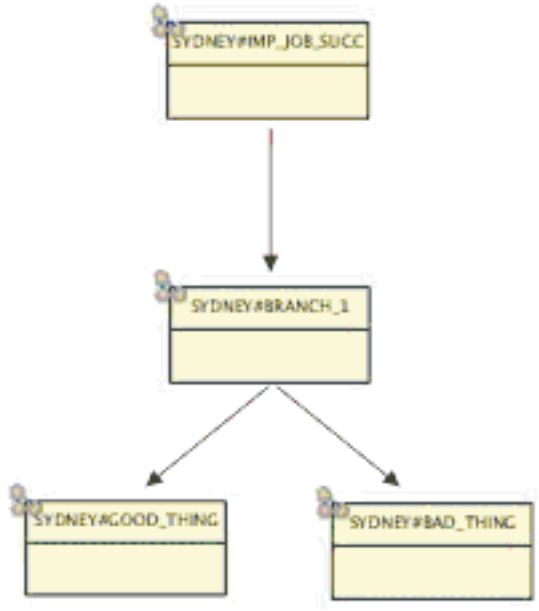


Figura 42. Definição de ramificação simples (SUCC)

Se a tarefa-pai for encerrada no status SUCC, o filho inválido será cancelado e o filho válido será executado. A Figura 43 mostra o estado final do fluxo de tarefas no plano atual.

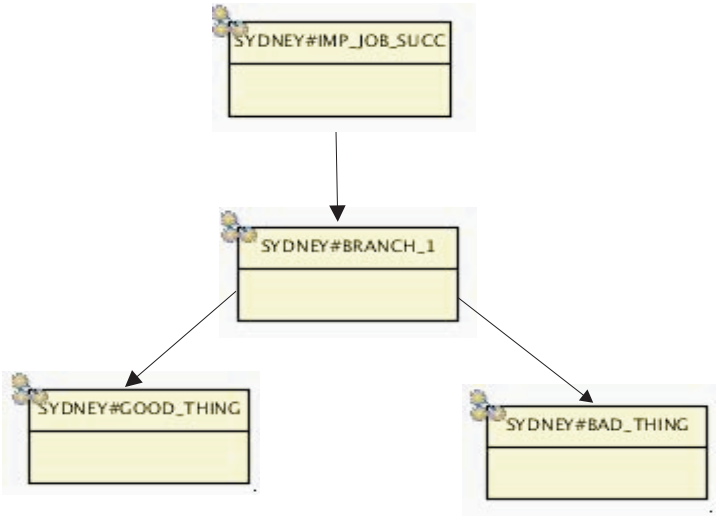


Figura 43. Status final de ramificação simples (SUCC)

Encerramento de Ramificação Simples no Estado ABEND

A Figura 44 na página 735 mostra a definição de fluxo de tarefas.

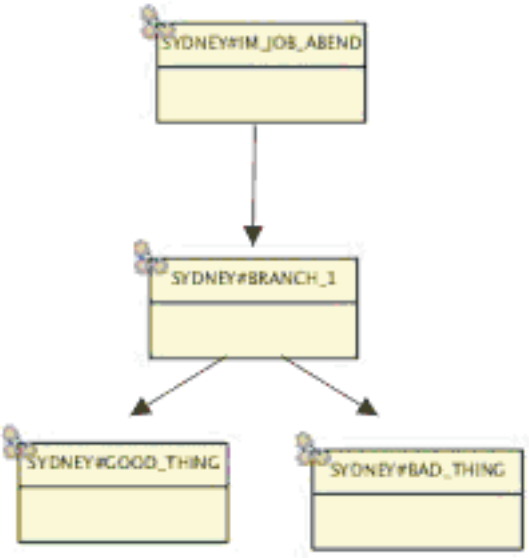


Figura 44. Definição de ramificação simples (ABEND)

A tarefa avaliada (ou seja, o pai) deve ter a Opção de recuperação configurada como `Continue`, caso contrário, a tarefa ENCERRADA DE FORMA ANORMAL não liberará a tarefa de ramificação da dependência FOLLOWS e o fluxo de tarefas inteiro será encerrado no estado STUCK. Isto é válido para *todas* as tarefas-pai para as quais você avalia o estado do resultado, porque é necessário considerar que qualquer tarefa pode ser encerrada de forma anormal.

O exemplo a seguir mostra o log da instância da tarefa de ramificação genérica.

```
============ START of branch job ===================
=============== Job environment ====================
MASTER_PLATFORM=UNIX
STREAM_NAME=SIMPLE_BRANCH_B
PARAMETER_PREFIX=
JOB_NAME=BRANCH_1
BRANCH_SUFFIX=1
PARENT=IMPORTANT_JOB_ABEND
====================================================
============= Input parameters =============
INPUT_SWITCH=PARENT_SUCCESS
ACTION_SWITCH=CANCEL
CONDITION_COUNT=0
====================================================
============= MAIN DECISION MAKING =================
Evaluation dependent on PARENT_SUCCESS
FALSE: Searched for SUCC parent.
Status of PARENT JOB(IMPORTANT_JOB_ABEND) is ABEND.
====================================================
============= Action on STOP Branch ===============
Performing action CANCEL on job SYDNEY#0AAAAAAAAAAAAEC4.G_DO_THE_GOOD_THING
%cj SYDNEY#0AAAAAAAAAAAAEC4.G_DO_THE_GOOD_THING;schedid;noask
Command forwarded to batchman for SYDNEY#GBJ_LONG_ABEND[(2159 12/16/07),
   (0AAAAAAAAAAAAEC4)].G_DO_THE_GOOD_THING
Performing action CANCEL on job SYDNEY#0AAAAAAAAAAAAEC4.SOME_GOOD_1
%cj SYDNEY#0AAAAAAAAAAAAEC4.SOME_GOOD_1;schedid;noask
Command forwarded to batchman for SYDNEY#GBJ_LONG_ABEND[(2159 12/16/07),
   (0AAAAAAAAAAAAEC4)].SOME_GOOD_1
Performing action CANCEL on job SYDNEY#0AAAAAAAAAAAAEC4.SOME_GOOD_2
%cj SYDNEY#0AAAAAAAAAAAAEC4.SOME_GOOD_2;schedid;noask
Command forwarded to batchman for SYDNEY#GBJ_LONG_ABEND[(2159 12/16/07),
   (0AAAAAAAAAAAAEC4)].SOME_GOOD_2
====================================================
============= Action on RUN Branch =================
Performing action RELEASE on job SYDNEY#0AAAAAAAAAAAAEC4.B_DO_THE_BAD_THING
Releasing of job SYDNEY#0AAAAAAAAAAAAEC4.B_DO_THE_BAD_THING is NOT NECESSARY,
   because priority=10
====================================================
============= Statistics of branch job BRANCH_1 =============
FALSE: Searched for SUCC parent. Status of PARENT JOB(IMPORTANT_JOB_ABEND) is ABEND.
For action CANCEL-RUN_BRANCH=B_DO_THE_BAD_THING and STOP_BRANCH=G_DO_THE_GOOD_THING
BRANCH selected to STOP: G_DO_THE_GOOD_THING
BRANCH selected to CONTINUE: B_DO_THE_BAD_THING
CANCELED_JOBS: G_DO_THE_GOOD_THING, SOME_GOOD_1, SOME_GOOD_2
PAUSED_JOB:
RELEASED_JOB:
============ END of branch job BRANCH_1 ==========
```

Parâmetros de Entrada Necessários

A ramificação baseada na avaliação do status da tarefa-pai é o uso mais comum da tarefa de ramificação genérica; ela *não* requer parâmetros de entrada.

Colocando a Tarefa de Ramificação no Fluxo de Tarefas

Coloque a tarefa de ramificação genérica no fluxo de tarefas após a tarefa-pai e renomeie o filho válido com o prefixo "G_" e o filho inválido com o prefixo "B_".

Além disso, siga a melhor prática e renomeie a tarefa de ramificação com um sufixo que consiste no caractere de sublinhado e em um valor numérico. Um nome típico para a primeira tarefa de ramificação em um fluxo de tarefas é BRANCH_1.

Cenário de Ramificação Longa

Ramificação longa é o uso recursivo de ramificação simples.

Uso de Ramificação Longa

O principal propósito do cenário de ramificação longa é mostrar que a tarefa de ramificação genérica pode cancelar *todas* as tarefas na ramificação de parada, mesmo que exista uma árvore inteira de tarefas para cancelar.

Esta função é necessária porque, no Tivoli Workload Scheduler, se uma tarefa for cancelada, todos os sucessores da tarefa serão liberados da dependência FOLLOWS, com o resultado indicando que as tarefas que são *dependentes* da tarefa *cancelada* serão iniciadas imediatamente. Como este comportamento pode ser indesejado em alguns casos, a tarefa de ramificação genérica cancelará a primeira tarefa de parada e *todos* os seus sucessores.

Nota: De um ponto de vista de programação, a tarefa de ramificação genérica usa chamadas de função recursivas para passar por todos os sucessores do primeiro filho de parada da tarefa de ramificação genérica.

Encerramento de Ramificação Longa no Estado SUCC

A Figura 45 na página 737 mostra o fluxo de tarefas que contém a estrutura complexa de possíveis sucessores, na ramificação válida ou na inválida.

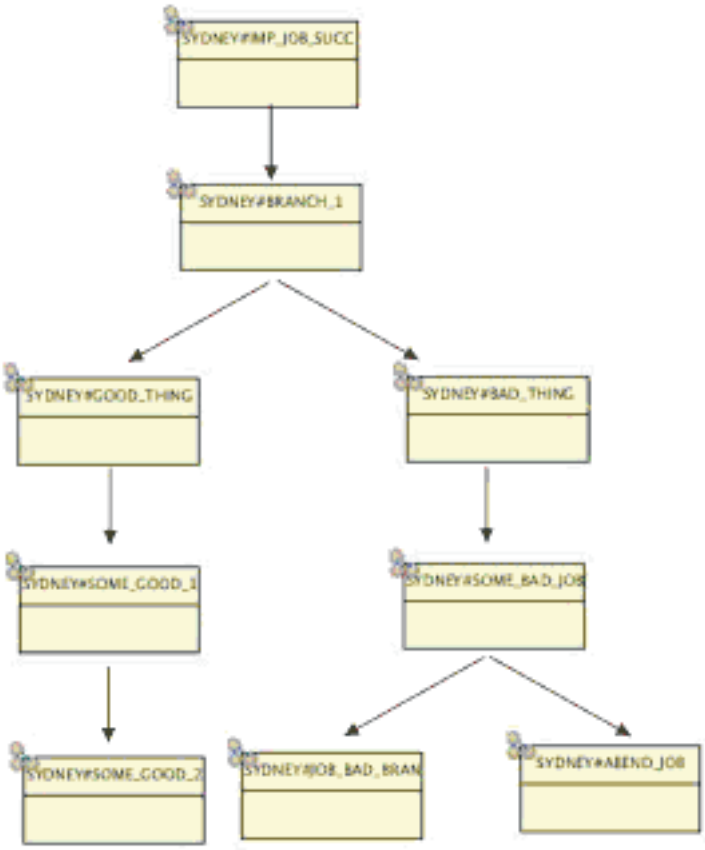


Figura 45. Definição de ramificação longa (SUCC)

Se o pai foi encerrado no estado SUCC, o log da tarefa mostrará a seguinte saída:

```
============ START of branch job BRANCH_1 ==========
====================== Job environment =============
MASTER_PLATFORM=UNIX
STREAM_NAME=GBJ_LONG_SUCC
STREAM_CPU=SYDNEY
BRANCH_JOB_NAME=BRANCH_1
PARENT=IMPORTANT_JOB_SUCC
============================================================
============= Input parameters =============
CONDITION_SWITCH=PARENT_SUCCESS
ACTION_SWITCH=CANCEL
============================================================
============= MAIN DECISION MAKING =============
Evaluation dependent on PARENT_SUCCESS
TRUE: Searched for SUCC parent. Status of PARENT JOB(IMPORTANT_JOB_SUCC) is SUCC
============================================================
============= Action on STOP Branch =============
Performing action CANCEL on job SYDNEY#0AAAAAAAAAAAAEC3.B_DO_THE_BAD_THING
%cj SYDNEY#0AAAAAAAAAAAAEC3.B_DO_THE_BAD_THING;schedid;noask
Command forwarded to batchman for SYDNEY#GBJ_LONG_SUCC[(2154 12/16/07),
   (0AAAAAAAAAAAAEC3)].B_DO_THE_BAD_THING
Performing action CANCEL on job SYDNEY#0AAAAAAAAAAAAEC3.SOME_BAD_JOB
%cj SYDNEY#0AAAAAAAAAAAAEC3.SOME_BAD_JOB;schedid;noask
Command forwarded to batchman for SYDNEY#GBJ_LONG_SUCC[(2154 12/16/07),
   (0AAAAAAAAAAAAEC3)].SOME_BAD_JOB
Performing action CANCEL on job SYDNEY#0AAAAAAAAAAAAEC3.ABEND_JOB
%cj SYDNEY#0AAAAAAAAAAAAEC3.ABEND_JOB;schedid;noask
Command forwarded to batchman for SYDNEY#GBJ_LONG_SUCC[(2154 12/16/07),
   (0AAAAAAAAAAAAEC3)].ABEND_JOB
Performing action CANCEL on job SYDNEY#0AAAAAAAAAAAAEC3.ANOTHER_JOB_IN_BAD_BRANCH
%cj SYDNEY#0AAAAAAAAAAAAEC3.ANOTHER_JOB_IN_BAD_BRANCH;schedid;noask
Command forwarded to batchman for SYDNEY#GBJ_LONG_SUCC[(2154 12/16/07),
   (0AAAAAAAAAAAAEC3)].ANOTHER_JOB_IN_BAD_BRANCH
============================================================
============= Action on RUN Branch =============
Performing action RELEASE on job SYDNEY#0AAAAAAAAAAAAEC3.G_DO_THE_GOOD_THING
Releasing of job SYDNEY#0AAAAAAAAAAAAEC3.G_DO_THE_GOOD_THING is NOT NECESSARY,
   because priority=10
============================================================
============= Statistics of branch job BRANCH_1 =============
TRUE: Searched for SUCC parent. Status of PARENT JOB(IMPORTANT_JOB_SUCC) is SUCC.
For action CANCEL-RUN_BRANCH=G_DO_THE_GOOD_THING and STOP_BRANCH=B_DO_THE_BAD_THING
BRANCH selected to STOP: B_DO_THE_BAD_THING
BRANCH selected to CONTINUE: G_DO_THE_GOOD_THING
CANCELED_JOBS: B_DO_THE_BAD_THING, SOME_BAD_JOB, ABEND_JOB, ANOTHER_JOB_IN_BAD_BRANCH
PAUSED_JOB:
RELEASED_JOB:
============ END of branch job BRANCH_1 ==========
```

Encerramento de Ramificação Longa no Estado ABEND

A Figura 46 mostra como o fluxo de tarefas será executado se a tarefa avaliada for encerrada de forma anormal.

Nota: Para liberar os sucessores da dependência FOLLOWS, a tarefa avaliada (tarefa-pai) deverá ter a opção de recuperação configurada como `Continue`. É possível configurar este parâmetro apenas na definição de tarefa, não na definição de fluxo de tarefas.

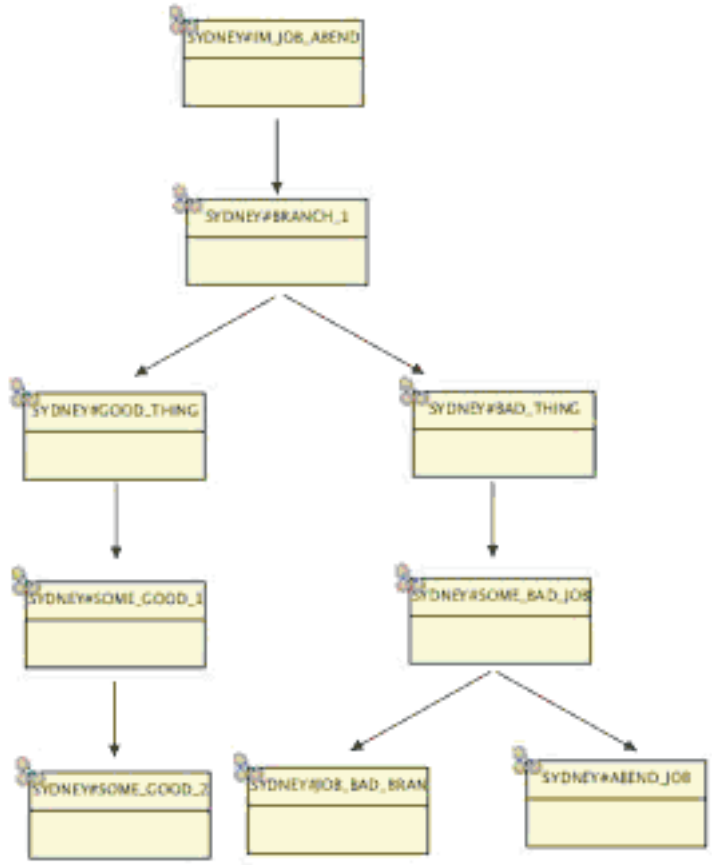


Figura 46. Status final (ABEND) da ramificação longa

O log da tarefa mostra a saída da instância da tarefa de ramificação genérica:

```
============ START of branch job BRANCH_1 ==========
====================== Job environment =============
MASTER_PLATFORM=UNIX
STREAM_NAME=GBJ_LONG_ABEND
STREAM_CPU=SYDNEY
BRANCH_JOB_NAME=BRANCH_1
PARENT=IMPORTANT_JOB_ABEND
============================================================
============= Input parameters =============
CONDITION_SWITCH=PARENT_SUCCESS
ACTION_SWITCH=CANCEL
============================================================
============= MAIN DECISION MAKING =============
Evaluation dependent on PARENT_SUCCESS
FALSE: Searched for SUCC parent.
Status of PARENT JOB(IMPORTANT_JOB_ABEND) is ABEND.
============================================================
============= Action on STOP Branch =============
Performing action CANCEL on job SYDNEY#0AAAAAAAAAAAEC4.G_DO_THE_GOOD_THING
%cj SYDNEY#0AAAAAAAAAAAEC4.G_DO_THE_GOOD_THING;schedid;noask
Command forwarded to batchman for SYDNEY#GBJ_LONG_ABEND[(2159 12/16/07),
   (0AAAAAAAAAAAEC4)].G_DO_THE_GOOD_THING
Performing action CANCEL on job SYDNEY#0AAAAAAAAAAAEC4.SOME_GOOD_1
%cj SYDNEY#0AAAAAAAAAAAEC4.SOME_GOOD_1;schedid;noask
Command forwarded to batchman for SYDNEY#GBJ_LONG_ABEND[(2159 12/16/07),
   (0AAAAAAAAAAAEC4)].SOME_GOOD_1
Performing action CANCEL on job SYDNEY#0AAAAAAAAAAAEC4.SOME_GOOD_2
%cj SYDNEY#0AAAAAAAAAAAEC4.SOME_GOOD_2;schedid;noask
Command forwarded to batchman for SYDNEY#GBJ_LONG_ABEND[(2159 12/16/07),
```

```
   (0AAAAAAAAAAAEC4)].SOME_GOOD_2
============================================================
============= Action on RUN Branch =============
Performing action RELEASE on job SYDNEY#0AAAAAAAAAAAEC4.B_DO_THE_BAD_THING
Releasing of job SYDNEY#0AAAAAAAAAAAEC4.B_DO_THE_BAD_THING is NOT NECESSARY,
   because priority=10
============================================================
============= Statistics of branch job BRANCH_1 =============
FALSE: Searched for SUCC parent. Status of PARENT JOB(IMPORTANT_JOB_ABEND) is ABEND.
For action CANCEL-RUN_BRANCH=B_DO_THE_BAD_THING and STOP_BRANCH=G_DO_THE_GOOD_THING
BRANCH selected to STOP: G_DO_THE_GOOD_THING
BRANCH selected to CONTINUE: B_DO_THE_BAD_THING
CANCELED_JOBS: G_DO_THE_GOOD_THING, SOME_GOOD_1, SOME_GOOD_2
PAUSED_JOB:
RELEASED_JOB:
============ END of branch job BRANCH_1 ==========
```

Parâmetros de Entrada Necessários

Esse tipo de ramificação não requer nenhum parâmetro de entrada.

Colocando a Tarefa de Ramificação no Fluxo de Tarefas

Coloque a tarefa de ramificação genérica no fluxo de tarefas após a tarefa-pai e renomeie o filho válido com o prefixo "G_" e o filho inválido com o prefixo "B_".

Além disso, siga a melhor prática e renomeie a tarefa de ramificação com um sufixo que consiste no caractere de sublinhado e em um valor numérico. Um nome típico para a primeira tarefa de ramificação em um fluxo de tarefas é BRANCH_1.

Diversas Ramificações

Use diversas ramificações para ter diversas tarefas de ramificação diferentes em um único fluxo de tarefas.

Uso de Diversas Ramificações

O propósito de ter mais de uma tarefa de ramificação em um único fluxo de tarefas é executar a ramificação várias vezes. A Figura 47 na página 740 mostra um exemplo de um fluxo de tarefas gerenciado por diversas tarefas de ramificação.

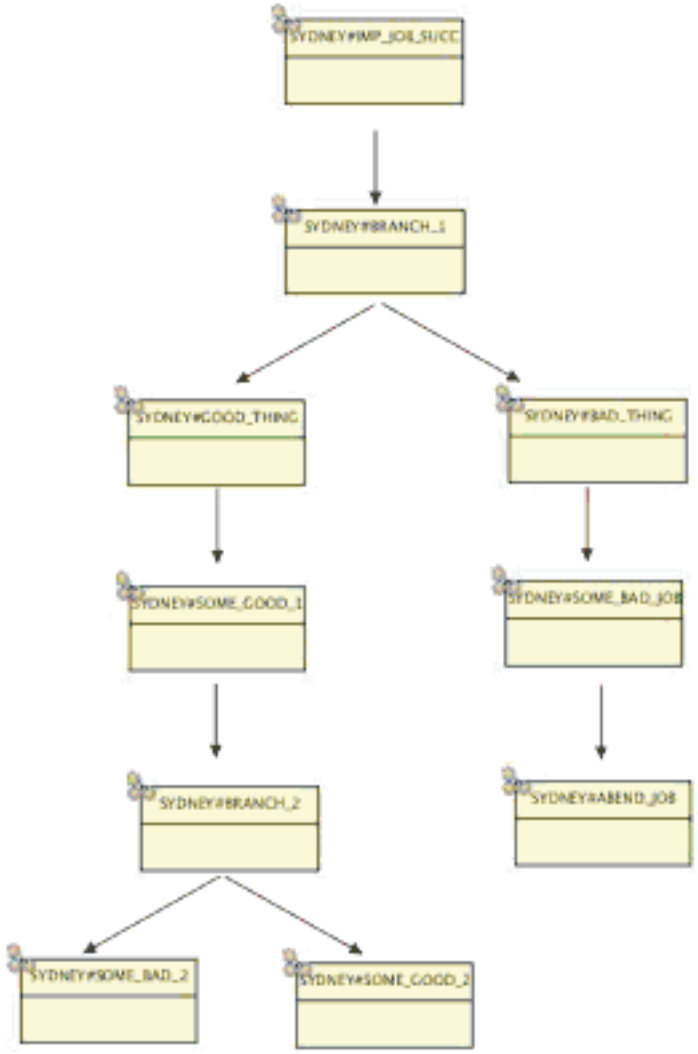


Figura 47. Diversas Tarefas de Ramificação em um Fluxo de Tarefas

Parâmetros de Entrada Necessários

Pode ser necessário especificar parâmetros de entrada, dependendo das tarefas de ramificação usadas. Não é necessário especificar parâmetros para tarefas com base nos seguintes cenários:

- Ramificação simples
- Ramificação longa

Especifique parâmetros de entrada para tarefas com base nos seguintes cenários:

- Ramificação Complexa - Padrão
- Ramificação Complexa - Padrão na Linha de Padrão
- Ramificação Complexa - Comparação de Valor Numérico
- Cenário complexo - diversas condições
- Cenário de Ações Pausar/Liberar
- Cenário de Ação de Sinal

Colocando as Tarefas de Ramificação no Fluxo de Tarefas

Para cada tarefa de ramificação, coloque a tarefa de ramificação genérica no fluxo de tarefas após a tarefa-pai e renomeie o filho válido com o prefixo "G_" e o filho inválido com o prefixo "B_". Faça a distinção entre as diferentes tarefas de ramificação por *sufixo de ramificação*, que deve conter um *caractere de sublinhado* e um *número*.

A melhor prática é criar o sufixo de ramificação usando números em ordem crescente, para que diversas tarefas de ramificação em um fluxo de tarefas sejam nomeadas, por exemplo, BRANCH_1, BRANCH_2, e assim por diante.

Encerramento Anormal do Pai

O cenário de encerramento anormal do pai representa o caso invertido do cenário de ramificação simples.

Uso do Encerramento Anormal do Pai

O cenário de encerramento anormal do pai descreve os seguintes requisitos funcionais:

- Obter o status da tarefa-pai.
- Se o status da tarefa-pai for SUCC, este será considerado como BAD.
- Se o status da tarefa-pai for ABEND, este será considerado como GOOD.

Esta seção descreve apenas o caso em que a tarefa-pai é encerrada no estado SUCC.

Encerramento Anormal do Pai no Estado SUCC

A Figura 48 mostra o fluxo de tarefas com a função PARENT_ABEND.

O fluxo de tarefas parece a ramificação simples, a única diferença é que você especificou o parâmetro CONDITION_SWITCH=PARENT_ABEND para a tarefa de ramificação.

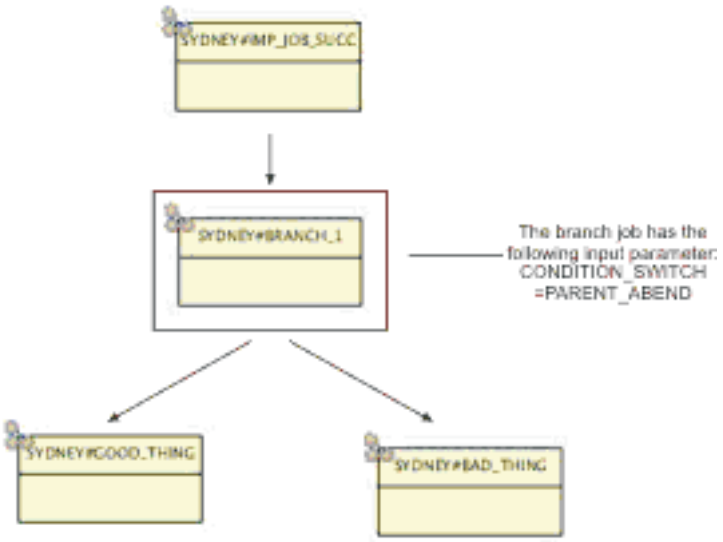


Figura 48. Definição de encerramento anormal do pai (SUCC)

O log da tarefa mostra a saída da instância da tarefa de ramificação genérica:

```
============ START of branch job BRANCH_1 ==========
================ Job environment ====================
MASTER_PLATFORM=UNIX
STREAM_NAME=GBJ_PARENT_ABEND
STREAM_CPU=SYDNEY
BRANCH_JOB_NAME=BRANCH_1
PARENT=IMPORTANT_JOB_SUCC
============================================================
============= Input parameters =============
CONDITION_SWITCH=PARENT_ABEND
ACTION_SWITCH=CANCEL
============================================================
============= MAIN DECISION MAKING =============
Evaluation dependent on PARENT_ABEND
FALSE: Searched for ABEND parent.
Status of PARENT JOB(IMPORTANT_JOB_SUCC) is SUCC.
============================================================
============= Action on STOP Branch =============
Performing action CANCEL on job SYDNEY#0AAAAAAAAAAAAEDA.G_DO_THE_GOOD_THING
%cj SYDNEY#0AAAAAAAAAAAAEDA.G_DO_THE_GOOD_THING;schedid;noask
Command forwarded to batchman for SYDNEY#GBJ_PARENT_ABEND[(2314 12/16/07),
   (0AAAAAAAAAAAAEDA)].G_DO_THE_GOOD_THING
============================================================
============= Action on RUN Branch =============
Performing action RELEASE on job SYDNEY#0AAAAAAAAAAAAEDA.B_DO_THE_BAD_THING
Releasing of job SYDNEY#0AAAAAAAAAAAAEDA.B_DO_THE_BAD_THING is NOT NECESSARY,
because priority=10
============================================================
============= Statistics of branch job BRANCH_1 =============
FALSE: Searched for ABEND parent. Status of PARENT JOB(IMPORTANT_JOB_SUCC) is SUCC
For action CANCEL - RUN_BRANCH=B_DO_THE_BAD_THING and
   STOP_BRANCH=G_DO_THE_GOOD_THING
BRANCH selected to STOP: G_DO_THE_GOOD_THING
BRANCH selected to CONTINUE: B_DO_THE_BAD_THING
CANCELED_JOBS: G_DO_THE_GOOD_THING
PAUSED_JOB:
RELEASED_JOB:
============ END of branch job BRANCH_1 ==========
```

Parâmetros de Entrada Necessários

A Tabela 122 mostra os parâmetros necessários para o cenário de ramificação negativa.

Tabela 122. Parâmetros de entrada para o cenário de tarefa de ramificação negativa

Nome do parâmetro	Valor de parâmetro
CONDITION_SWITCH	PARENT_ABEND

O exemplo a seguir mostra a definição de parâmetro. O texto é inserido no campo Comentários da definição de fluxo de tarefas.

```
BRANCH_1-BEGIN
CONDITION_SWITCH=PARENT_ABEND
BRANCH_1-END
```

Para obter detalhes sobre como especificar os parâmetros da tarefa de ramificação, consulte “Especificando os Parâmetros da Tarefa de Ramificação” na página 770.

Colocando a Tarefa de Ramificação no Fluxo de Tarefas

Coloque a tarefa de ramificação genérica no fluxo de tarefas após a tarefa-pai e renomeie o filho válido com o prefixo "G_" e o filho inválido com o prefixo "B_".

Além disso, siga a melhor prática e renomeie a tarefa de ramificação com um sufixo que consiste no caractere de sublinhado e em um valor numérico. Um nome típico para a primeira tarefa de ramificação em um fluxo de tarefas é BRANCH_1.

Cenários de Ramificação Complexa

Os cenários de ramificação complexa mostram a flexibilidade da tarefa de ramificação genérica.

Os cenários anteriores eram baseados na avaliação do *status* da tarefa-pai. Os cenários da tarefa de ramificação complexa são baseados na avaliação do *log da tarefa* da tarefa-pai. Para obter uma lista das condições complexas, consulte Terminologia.

Uma condição complexa pode incluir diversas subcondições. Por exemplo:

1. subcondição 1
   - Procure o padrão de texto (fornecido como um parâmetro) no log da tarefa-pai.
   - Na linha em que o padrão está localizado, isole o valor numérico.
   - Compare o valor numérico com um número especificado. Para essa comparação, use o operador aritmético especificado (fornecido como um parâmetro). Se a comparação aritmética for bem-sucedida, ela retornará TRUE.

   Por exemplo, procure a linha que contém o padrão `Disk free`. Se localizá-la, procure um número na linha. Se o número for maior que 90, retorne TRUE.
2. subcondição 2
   a. Procure outro padrão de texto no log da tarefa do pai.
   b. Se o padrão for localizado, negue o resultado e retorne FALSE (por exemplo, retorne FALSE se o log da tarefa-pai contiver o padrão `Error`).
3. Conecte o primeiro e o segundo resultados da subcondição a um operador booleano (o operador booleano é fornecido como um parâmetro e os valores possíveis são AND e OR).
4. Avalie o resultado final da condição inteira (TRUE ou FALSE).

A condição definida pode ser muito flexível e pode abranger muitas situações típicas.

As seções a seguir descrevem vários cenários de ramificação complexa para explicar como:

- Procurar um padrão de texto específico no log da tarefa do pai.
- Procurar um padrão de texto específico no log da tarefa do pai. Se o padrão for localizado, procure um padrão de texto adicional na mesma linha.
- Procurar um padrão de texto específico no log da tarefa do pai. Se o padrão for localizado, procure um valor numérico na mesma linha. Se o valor for localizado, compare-o com o número fornecido usando o operador aritmético fornecido.
- Associar várias condições em uma usando os operadores booleanos AND e OR.

Ramificação Complexa - Padrão

Use o cenário de padrão de procura complexa para procurar um padrão de texto específico no log da tarefa-pai.

Uso de Cenário de Padrão

É possível procurar qualquer sequência, por exemplo, `ended successfully` ou `mounted ALL tape drives`. Em geral, o texto deve representar a mensagem positiva incluída no log da tarefa-pai.

Nota: A procura de padrão geralmente procura a mensagem positiva no log da tarefa-pai. Em alguns casos, talvez você queira implementar a lógica reversa, por exemplo, procurar o padrão `Error` , `Unsuccessfully` ou `Not Enough Space`, que representa mensagens negativas do log. Para usar esta abordagem, consulte “Ramificação Complexa - Padrão Negado” na página 745.

Usando a ramificação de padrão, se o padrão de texto for localizado, CONDITION=TRUE, caso contrário, CONDITION=FALSE.

A Figura 49 mostra a definição de fluxo de tarefas para o cenário de ramificação de padrão. O fluxo de tarefas parece semelhante ao cenário de ramificação simples, mas os parâmetros adicionais são definidos:

- CONDITION_SWITCH=COMPLEX
- PATTERN_1=completed successfully

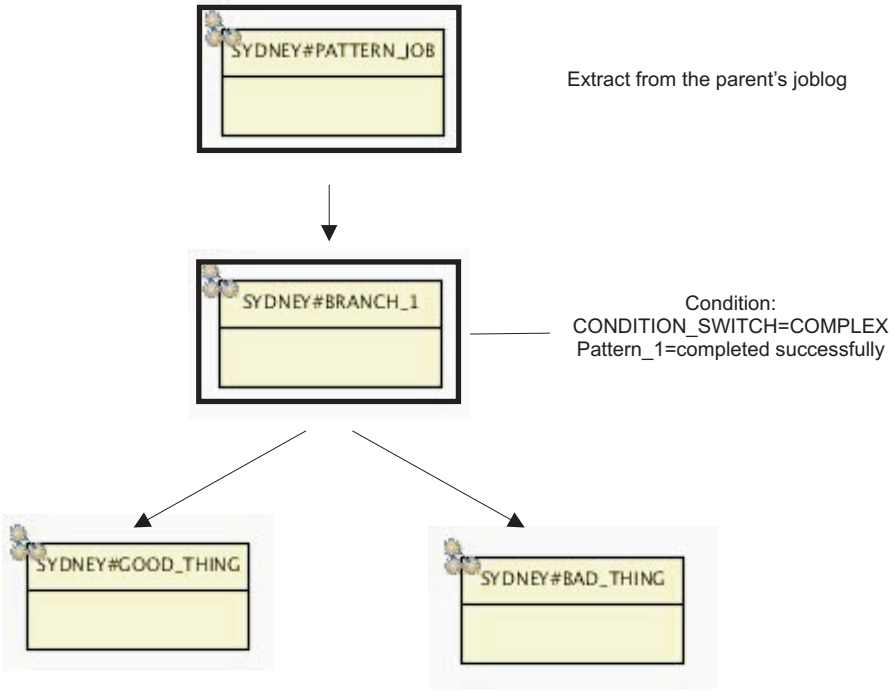


Figura 49. Cenário de Padrão - Definição

O log da tarefa do cenário de padrão mostra a saída da instância da tarefa de ramificação genérica:

```
============ START of branch job BRANCH_1 ==========
================== Job environment =================
MASTER_PLATFORM=UNIX
STREAM_NAME=GBJ_PATTERN
STREAM_CPU=SYDNEY
BRANCH_JOB_NAME=BRANCH_1
PARENT=PATTERN_JOB
============================================================
============= Input parameters =============
CONDITION_SWITCH=COMPLEX
ACTION_SWITCH=CANCEL
CONDITION_COUNT=1
PATTERN[1]=completed successfully
IS_CASE_SENSITIVE[1]=YES
IS_REGULAR_EXPRESSION[1]=NO
NEGATE_CONDITION_RESULT[1]=NO
============================================================
============= MAIN DECISION MAKING =============
COMPLEX condition evaluation
----------ATOMIC CONDITION 1------------
Searching for "completed successfully" in JOBLOG of PATTERN_JOB
Pattern FOUND, performing further tests.
No additional value defined for specified pattern.
Condition evaluated as TRUE.
ATOMIC CONDITION RESULT [1]= TRUE
-------------------------------------
-------- COMPLEX CONDITION ----------
[ TRUE ]
CONDITION_RESULT=TRUE
TRUE: The result of complex condition is TRUE.
============================================================
============= Action on STOP Branch =============
Performing action CANCEL on job SYDNEY#0AAAAAAAAAAAED4.B_DO_THE_BAD_THING
%cj SYDNEY#0AAAAAAAAAAAED4.B_DO_THE_BAD_THING;schedid;noask
Command forwarded to batchman for SYDNEY#GBJ_PATTERN[(1733 12/17/07),
   (0AAAAAAAAAAAED4)].B_DO_THE_BAD_THING
============================================================
============= Action on RUN Branch =============
Performing action RELEASE on job SYDNEY#0AAAAAAAAAAAED4.G_DO_THE_GOOD_THING
Releasing of job SYDNEY#0AAAAAAAAAAAED4.G_DO_THE_GOOD_THING is NOT NECESSARY,
   because priority=10
============================================================
============= Statistics of branch job BRANCH_1 =============
TRUE: The result of complex condition is TRUE.
```

```
For action CANCEL - RUN_BRANCH=G_DO_THE_GOOD_THING and
   STOP_BRANCH=B_DO_THE_BAD_THING
BRANCH selected to STOP: B_DO_THE_BAD_THING
BRANCH selected to CONTINUE: G_DO_THE_GOOD_THING
CANCELED_JOBS: B_DO_THE_BAD_THING
PAUSED_JOB:
RELEASED_JOB:
============ END of branch job BRANCH_1 ==========
```

Parâmetros de Entrada Necessários

A Tabela 123 mostra os parâmetros necessários para o cenário de padrão.

Tabela 123. Parâmetros de entrada para o cenário da tarefa de padrão

Nome do parâmetro	Valor de parâmetro
CONDITION_SWITCH	COMPLEX
PATTERN_1	Completed successfully

O exemplo a seguir mostra a definição de parâmetro. O texto é inserido no campo Comentários da definição de fluxo de tarefas.

```
BRANCH_1-BEGIN
CONDITION_SWITCH=COMPLEX
PATTERN_1=completed successfully
BRANCH_1-END
```

Para obter uma descrição sobre como especificar os parâmetros para a tarefa de ramificação, consulte Trabalhando com Parâmetros da Tarefa de Ramificação.

Colocando a Tarefa de Ramificação no Fluxo de Tarefas

Coloque a tarefa de ramificação genérica no fluxo de tarefas após a tarefa-pai e renomeie o filho válido com o prefixo "G_" e o filho inválido com o prefixo "B_".

Além disso, siga a melhor prática e renomeie a tarefa de ramificação com um sufixo que consiste no caractere de sublinhado e em um valor numérico. Um nome típico para a primeira tarefa de ramificação em um fluxo de tarefas é BRANCH_1.

Ramificação Complexa - Padrão Negado

Este é o caso invertido do cenário de ramificação de padrão.

Uso do Cenário de Padrão Negado

Com o cenário de ramificação de padrão, você procura no log da tarefa-pai um padrão considerado como uma mensagem positiva (por exemplo, `completed successfully`). Se, em vez disso, você desejar procurar uma mensagem negativa (por exemplo, `Error`), use a tarefa de ramificação genérica para negar cada subcondição definida da condição complexa. Neste cenário, localizar os resultados do padrão como CONDITION=FALSE. Não localizar os resultados do padrão como CONDITION=TRUE.

A Figura 50 na página 746 mostra a definição de fluxo de tarefas para o cenário de ramificação de padrão com os seguintes parâmetros:

- CONDITION_SWITCH=COMPLEX
- PATTERN_1=Error
- NEGATE CONDITION_RESULT_1=YES

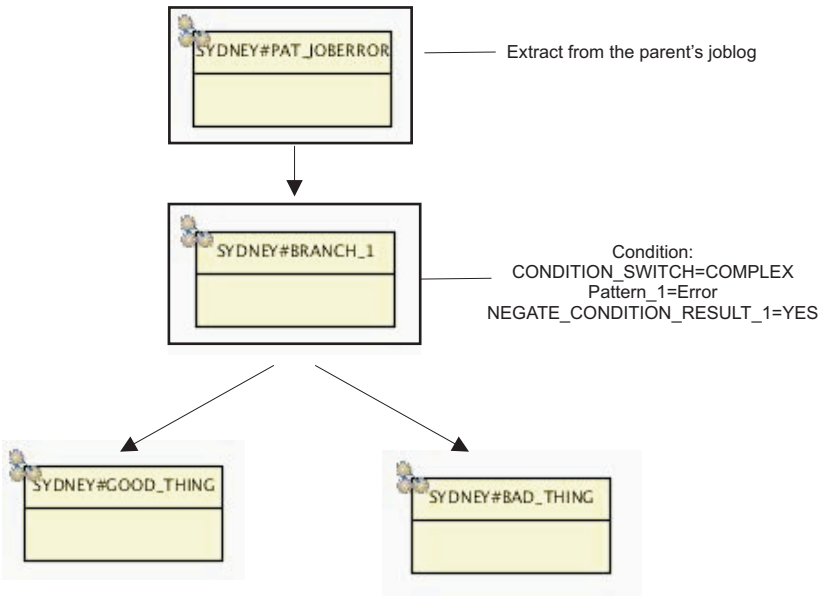


Figura 50. Definição de cenário de padrão negado

O log da tarefa mostra a saída da instância da tarefa de ramificação genérica:

```
============ START of branch job BRANCH_1 ==========
=================== Job environment ================
MASTER_PLATFORM=UNIX
STREAM_NAME=GBJ_PATTERN_NEG
STREAM_CPU=SYDNEY
BRANCH_JOB_NAME=BRANCH_1
PARENT=PATTERN_JOB_ERROR
============================================================
============= Input parameters =============
CONDITION_SWITCH=COMPLEX
ACTION_SWITCH=CANCEL
CONDITION_COUNT=1
PATTERN[1]=Error
IS_CASE_SENSITIVE[1]=YES
IS_REGULAR_EXPRESSION[1]=NO
NEGATE_CONDITION_RESULT[1]=YES
============================================================
============= MAIN DECISION MAKING =============
COMPLEX condition evaluation
----------ATOMIC CONDITION 1------------
Searching for "Error" in JOBLOG of PATTERN_JOB_ERROR
Pattern FOUND, performing further tests.
No additional value defined for specified pattern.
Condition evaluated as TRUE.
==NEGATED== ATOMIC CONDITION RESULT [1]= FALSE
------------------------------------
-------- COMPLEX CONDITION ----------
[ FALSE ]
CONDITION_RESULT=FALSE
FALSE: The result of complex condition is FALSE.
============================================================
============= Action on STOP Branch =============
Performing action CANCEL on job SYDNEY#0AAAAAAAAAAAAED3.G_DO_THE_GOOD_THING
%cj SYDNEY#0AAAAAAAAAAAAED3.G_DO_THE_GOOD_THING;schedid;noask
Command forwarded to batchman for SYDNEY#GBJ_PATTERN_NEG[(1730 12/17/07),
   (0AAAAAAAAAAAAED3)].G_DO_THE_GOOD_THING
============================================================
============= Action on RUN Branch =============
Performing action RELEASE on job SYDNEY#0AAAAAAAAAAAAED3.B_DO_THE_BAD_THING
Releasing of job SYDNEY#0AAAAAAAAAAAAED3.B_DO_THE_BAD_THING is NOT NECESSARY,
   because priority=10
============================================================
============= Statistics of branch job BRANCH_1 =============
FALSE: The result of complex condition is FALSE.
For action CANCEL - RUN_BRANCH=B_DO_THE_BAD_THING and
   STOP_BRANCH=G_DO_THE_GOOD_THING
BRANCH selected to STOP: G_DO_THE_GOOD_THING
BRANCH selected to CONTINUE: B_DO_THE_BAD_THING
CANCELED_JOBS: G_DO_THE_GOOD_THING
PAUSED_JOB:
RELEASED_JOB:
============ END of branch job BRANCH_1 ==========
```

Parâmetros de Entrada Necessários

A Tabela 124 na página 747 mostra os parâmetros necessários para o cenário de ramificação negada.

Tabela 124. Parâmetros de entrada para o cenário de tarefa de padrão negado

Nome do parâmetro	Valor de parâmetro
CONDITION_SWITCH	COMPLEX
PATTERN_1	Erro
NEGATE_CONDITION_RESULT_1	SIM

O exemplo a seguir mostra a definição de parâmetro. O texto é inserido no campo Comentários da definição de fluxo de tarefas.

```
BRANCH_1-BEGIN
CONDITION_SWITCH=COMPLEX
PATTERN_1=Error
NEGATE_CONDITION_1=YES
BRANCH_1-END
```

Para obter uma descrição sobre como especificar os parâmetros para a tarefa de ramificação, consulte “Especificando os Parâmetros da Tarefa de Ramificação” na página 770.

Colocando a Tarefa de Ramificação no Fluxo de Tarefas

Coloque a tarefa de ramificação genérica no fluxo de tarefas após a tarefa-pai e renomeie o filho válido com o prefixo "G_" e o filho inválido com o prefixo "B_".

Além disso, siga a melhor prática e renomeie a tarefa de ramificação com um sufixo que consiste no caractere de sublinhado e em um valor numérico. Um nome típico para a primeira tarefa de ramificação em um fluxo de tarefas é BRANCH_1.

Ramificação Complexa - Padrão na Linha de Padrão

Este cenário estende a função da procura de padrão.

Padrão no Uso de Cenário de Linha de Padrão

O propósito deste cenário é:

1. Obter o log da tarefa-pai.
2. No log da tarefa, identificar uma linha contendo um padrão específico.
3. Se a linha for localizada, procurar outro padrão na linha.
4. Se o segundo padrão for localizado, retornar a condição como TRUE.

O exemplo a seguir extraído de um log da tarefa-pai mostra como deve ser um cenário de uso típico:

```
Sending STOP signal to component XYZ: SUCCESS
Stopping component XYZ: SUCCESS
```

Com base nesse exemplo, uma procura simples pelo padrão de textoSUCCESS mostraria que o componente realmente tinha sido interrompido. Há diversas ocorrências do padrão SUCCESS e o uso da procura de padrão simples não determinaria o resultado correto.

Outro exemplo pode ser o seguinte resumo do log da tarefa:

```
About to stop component XYZ...
Sending STOP signal to component XYZ: SUCCESS
Stopping component XYZ: FAILED
```

Com base nesse exemplo, se você usou a procura de padrão simples, o resultado será CONDITION=TRUE, porque o padrão SUCCESS foi localizado. Mas, neste caso, o log da tarefa-pai não é avaliado corretamente.

Para avaliar corretamente o log da tarefa-pai, você deve usar o padrão no cenário de linha de padrão, da seguinte forma:

1. Procure o padrão `Stopping component`.
2. Se a linha for localizada, procure o padrão `SUCCESS`.
3. Se as duas procuras forem bem-sucedidas, retorne a condição como TRUE.

A Figura 51 mostra a definição para o cenário de ramificação padrão.

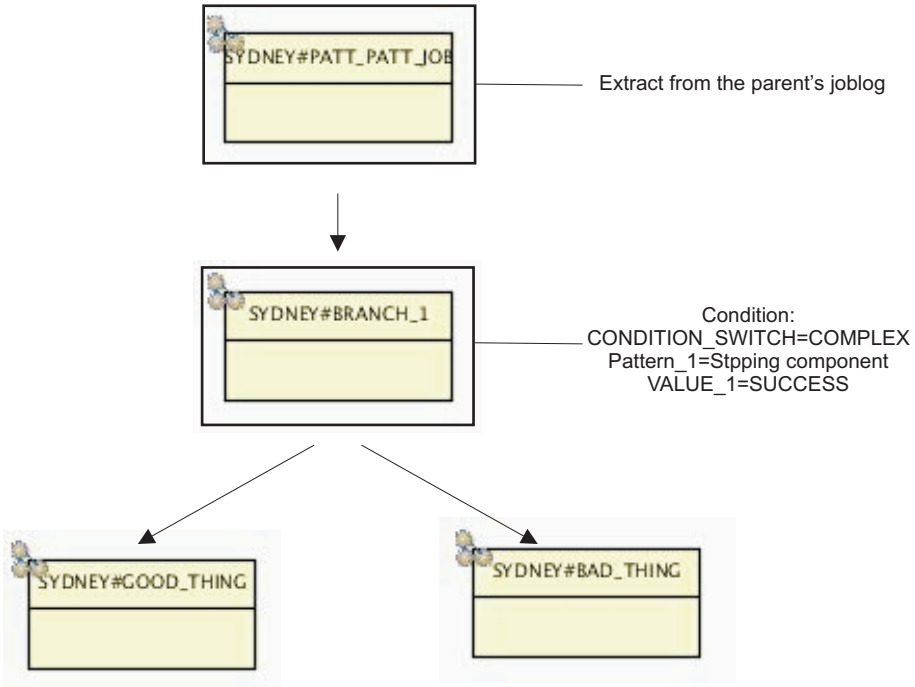


Figura 51. Padrão na Definição de Linha de Padrão

O log da tarefa mostra a saída da instância da tarefa de ramificação genérica:

```
============ START of branch job BRANCH_1 ==========
================= Job environment =================
MASTER_PLATFORM=UNIX
STREAM_NAME=GBJ_PATTERN_PATT
STREAM_CPU=SYDNEY
BRANCH_JOB_NAME=BRANCH_1
PARENT=PATTERN_PATTERN_JOB
============================================================
============= Input parameters =============
CONDITION_SWITCH=COMPLEX
ACTION_SWITCH=CANCEL
CONDITION_COUNT=1
PATTERN[1]=Stopping component
VALUE[1]=SUCCESS
IS_CASE_SENSITIVE[1]=YES
IS_REGULAR_EXPRESSION[1]=NO
NEGATE_CONDITION_RESULT[1]=NO
============================================================
============= MAIN DECISION MAKING =============
COMPLEX condition evaluation
----------ATOMIC CONDITION 1-----------
Searching for "Stopping component" in JOBLOG of PATTERN_PATTERN_JOB
Pattern FOUND, performing further tests.
Searching for STRING=SUCCESS within the
   row "Stopping component XYZ: FAILED".
String "SUCCESS" NOT found within the
   row "Stopping component XYZ: FAILED".
ATOMIC CONDITION RESULT [1]= FALSE
-------------------------------------
-------- COMPLEX CONDITION ----------
[ FALSE ]
CONDITION_RESULT=FALSE
FALSE: The result of complex condition is FALSE.
============================================================
============= Action on STOP Branch =============
Performing action CANCEL on job SYDNEY#0AAAAAAAAAAAED2.G_DO_THE_GOOD_THING
%cj SYDNEY#0AAAAAAAAAAAED2.G_DO_THE_GOOD_THING;schedid;noask
Command forwarded to batchman for SYDNEY#GBJ_PATTERN_PATT[(1634 12/17/07),
   (0AAAAAAAAAAAED2)].G_DO_THE_GOOD_THING
============================================================
============= Action on RUN Branch =============
```

```
Performing action RELEASE on job SYDNEY#0AAAAAAAAAAAED2.B_DO_THE_BAD_THING
Releasing of job SYDNEY#0AAAAAAAAAAAED2.B_DO_THE_BAD_THING is NOT NECESSARY,
   because priority=10
============================================================
============= Statistics of branch job BRANCH_1 =============
FALSE: The result of complex condition is FALSE.
For action CANCEL - RUN_BRANCH=B_DO_THE_BAD_THING and
   STOP_BRANCH=G_DO_THE_GOOD_THING
BRANCH selected to STOP: G_DO_THE_GOOD_THING
BRANCH selected to CONTINUE: B_DO_THE_BAD_THING
CANCELED_JOBS: G_DO_THE_GOOD_THING
PAUSED_JOB:
RELEASED_JOB:
============ END of branch job BRANCH_1 ==========
```

Parâmetros de Entrada Necessários

A Tabela 125 mostra os parâmetros necessários para o padrão no cenário de linha de padrão.

Tabela 125. Parâmetros de entrada para o padrão no cenário de linha de padrão

Nome do parâmetro	Valor de parâmetro
CONDITION_SWITCH	COMPLEX
PATTERN_1	Stopping component
VALUE_1	SUCCESS

A definição de parâmetro é semelhante ao seguinte exemplo. O texto é inserido no campo Comentários da definição de fluxo de tarefas.

```
BRANCH_1-BEGIN
CONDITION_SWITCH=COMPLEX
PATTERN_1=Stopping component
VALUE_1=SUCCESS
BRANCH_1-END
```

Para obter detalhes sobre como especificar os parâmetros da tarefa de ramificação, consulte “Especificando os Parâmetros da Tarefa de Ramificação” na página 770.

Colocando a Tarefa de Ramificação no Fluxo de Tarefas

Coloque a tarefa de ramificação genérica no fluxo de tarefas logo após a tarefa-pai e renomeie o filho válido com o prefixo "G_" e o filho inválido com o prefixo "B_".

Além disso, siga a melhor prática e renomeie a tarefa de ramificação com um sufixo que consiste no caractere de sublinhado e em um valor numérico. Um nome típico para a primeira tarefa de ramificação em um fluxo de tarefas é BRANCH_1.

Padrão na Linha de Padrão - Negado

Enquanto estiver no cenário de padrão negado, você nega o resultado de uma procura de padrão simples, no cenário de padrão negado na linha de padrão, você usa a abordagem *negada* para *qualquer* cenário complexo. É o caso inverso do cenário de padrão na linha de padrão.

Uso do Cenário de Padrão Negado na Linha de Padrão

Se você estiver executando uma procura de padrão simples, por exemplo, procurando a ocorrência `ERROR` no log da tarefa-pai, ou uma condição complexa, será possível usar a abordagem que nega o resultado.

Decida como analisar o log da tarefa-pai. Com base em seu entendimento do conteúdo do log da tarefa, escolha se a saída de linhas conterá uma mensagem positiva ou negativa. Com base neste conhecimento, decida se negará o resultado da subcondição específica ou o deixará como está.

A lógica de avaliação para o cenário de padrão negado na linha de padrão é a seguinte:

1. Procure a linha de padrão.
2. Procure o identificador principal, como `Backup on Primary device`.
3. Se a linha for localizada, procure a mensagem negativa `ERROR` nela.
4. Negue o resultado.

Esta abordagem cobre o cenário no qual você usa saídas como `SUCCESS`, `OK` e `COMPLETED` como mensagens positivas, e saídas como `FAILED` e `ERROR` como mensagens negativas.

A Figura 52 mostra a definição para o cenário de ramificação padrão.

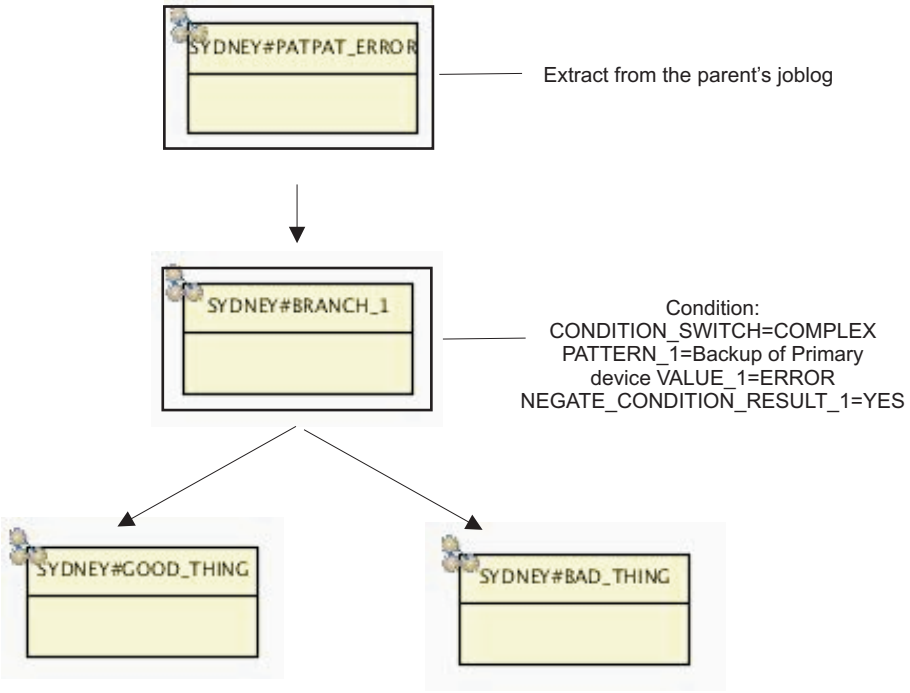


Figura 52. Definição de padrão na linha de padrão negada

Nota: Em *qualquer* cenário no qual você usa uma condição complexa, também é possível negar o resultado de *cada subcondição específica*. Também é possível configurar diversas subcondições atômicas e negar apenas algumas delas. Para obter informações detalhadas, consulte "Cenário Complexo - Diversas Condições" na página 754.

O log da tarefa mostra a saída da instância da tarefa de ramificação genérica:

```
============ START of branch job BRANCH_1 ==========
================= Job environment =================
MASTER_PLATFORM=UNIX
STREAM_NAME=GBJ_PAT_PAT_NEG
STREAM_CPU=SYDNEY
BRANCH_JOB_NAME=BRANCH_1
PARENT=PATTERN_PATTERN_ERROR_JOB
============================================================
============= Input parameters =============
CONDITION_SWITCH=COMPLEX
ACTION_SWITCH=CANCEL
CONDITION_COUNT=1
PATTERN[1]=Backup of Primary device
VALUE[1]=ERROR
IS_CASE_SENSITIVE[1]=YES
IS_REGULAR_EXPRESSION[1]=NO
NEGATE_CONDITION_RESULT[1]=YES
============================================================
============= MAIN DECISION MAKING =============
COMPLEX condition evaluation
----------ATOMIC CONDITION 1------------
Searching for "Backup of Primary device" in JOBLOG of
   PATTERN_PATTERN_ERROR_JOB
Pattern FOUND, performing further tests.
```

```
Searching for STRING=ERROR within the
   row "Backup of Primary device: ERROR".
String "ERROR" found within the
   row "Backup of Primary device: ERROR".
==NEGATED== ATOMIC CONDITION RESULT [1]= FALSE
--------------------------------------
-------- COMPLEX CONDITION ----------
[ FALSE ]
CONDITION_RESULT=FALSE
FALSE: The result of complex condition is FALSE.
============================================================
============= Action on STOP Branch =============
Performing action CANCEL on job SYDNEY#0AAAAAAAAAAAAEEA.G_DO_THE_GOOD_THING
%cj SYDNEY#0AAAAAAAAAAAAEEA.G_DO_THE_GOOD_THING;schedid;noask
Command forwarded to batchman for SYDNEY#GBJ_PAT_PAT_NEG[(1941 12/17/07),
   (0AAAAAAAAAAAAEEA)].G_DO_THE_GOOD_THING
============================================================
============= Action on RUN Branch =============
Performing action RELEASE on job SYDNEY#0AAAAAAAAAAAAEEA.B_DO_THE_BAD_THING
Releasing of job SYDNEY#0AAAAAAAAAAAAEEA.B_DO_THE_BAD_THING is NOT NECESSARY,
   because priority=10
============================================================
============= Statistics of branch job BRANCH_1 =============
FALSE: The result of complex condition is FALSE.
For action CANCEL - RUN_BRANCH=B_DO_THE_BAD_THING and
   STOP_BRANCH=G_DO_THE_GOOD_THING
BRANCH selected to STOP: G_DO_THE_GOOD_THING
BRANCH selected to CONTINUE: B_DO_THE_BAD_THING
CANCELED_JOBS: G_DO_THE_GOOD_THING
PAUSED_JOB:
RELEASED_JOB:
============ END of branch job BRANCH_1 ==========
```

Parâmetros de Entrada Necessários

A Tabela 126 mostra os parâmetros necessários para o cenário de padrão negado na linha de padrão.

Tabela 126. Parâmetros de entrada para o cenário de padrão negado na linha de padrão

Nome do parâmetro	Valor de parâmetro
CONDITION_SWITCH	COMPLEX
PATTERN_1	Backup of Primary device
VALUE_1	ERROR
NEGATE_CONDITION_RESULT_1	SIM

A definição de parâmetro é semelhante ao seguinte exemplo. O texto é inserido no campo Comentários da definição de fluxo de tarefas.

```
BRANCH_1-BEGIN
CONDITION_SWITCH=COMPLEX
PATTERN_1=Backup of Primary device
VALUE_1=ERROR
NEGATE_CONDITION_RESULT_1=YES
BRANCH_1-END
```

Para obter uma descrição sobre como especificar os parâmetros para a tarefa de ramificação, consulte “Especificando os Parâmetros da Tarefa de Ramificação” na página 770.

Colocando a Tarefa de Ramificação no Fluxo de Tarefas

Coloque a tarefa de ramificação genérica no fluxo de tarefas após a tarefa-pai e renomeie o filho válido com o prefixo "G_" e o filho inválido com o prefixo "B_".

Além disso, siga a melhor prática e renomeie a tarefa de ramificação com um sufixo que consiste no caractere de sublinhado e em um valor numérico. Um nome típico para a primeira tarefa de ramificação em um fluxo de tarefas é BRANCH_1.

Ramificação Complexa - Comparação de Valor Numérico

Para combinar a procura de padrão com a comparação de valor numérico.

Uso do Padrão na Ramificação de Linha de Padrão

Este cenário estende a função do cenário de padrão descrito em Ramificação Complexa - Padrão. O propósito deste cenário é:

1. Obter o log da tarefa-pai.
2. No log da tarefa, identifique a linha que contém um padrão específico.
3. Se a linha for localizada, procure um valor numérico nela.
4. Compare o número com o número fornecido como parâmetro de entrada, usando o operador aritmético que também é fornecido como um parâmetro.

O seguinte resumo do log da tarefa-pai mostra um cenário de uso típico:

```
Checking free space...
Free space on volume ABC is 50 %.
```

Você deseja executar a seguinte avaliação:

1. Procure o padrão `Free space on volume`.
2. Se o texto for localizado, tente extrair um valor numérico.
3. Compare o valor numérico da seguinte forma:

```
If (numeric_value > 30)
Then CONDITION=TRUE
Else CONDITION=FALSE
```

O número`30` e o operador `>` são parâmetros transmitidos para a tarefa de ramificação. É possível usar qualquer operador aritmético (por exemplo, `<`, `<=`, `>`). Para obter uma descrição sobre como especificar os parâmetros para a tarefa de ramificação, consulte “Especificando os Parâmetros da Tarefa de Ramificação” na página 770.

A Figura 53 na página 753 mostra a definição para o cenário de comparação de valor numérico. Também é possível reverter a lógica de avaliação, embora normalmente isso não seja necessário, porque é possível atingir o resultado de condição negado usando o operador oposto.

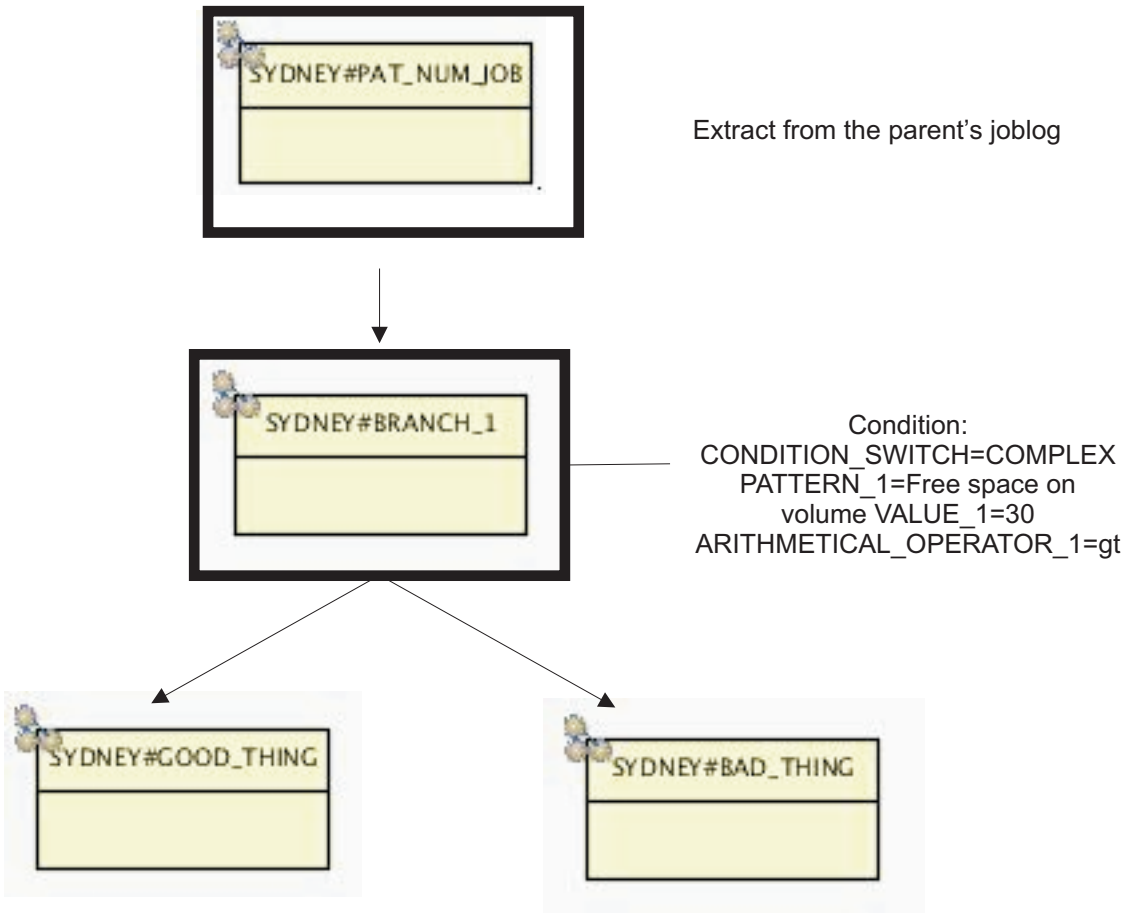


Figura 53. Definição de ramificação de comparação numérica

O log da tarefa mostra a saída da instância da tarefa de ramificação genérica:

```
============ START of branch job BRANCH_1 ==========
================ Job environment ==================
MASTER_PLATFORM=UNIX
STREAM_NAME=GBJ_PATTERN_NUM
STREAM_CPU=SYDNEY
BRANCH_JOB_NAME=BRANCH_1
PARENT=PATTERN_NUMBER_JOB
============================================================
============= Input parameters =============
CONDITION_SWITCH=COMPLEX
ACTION_SWITCH=CANCEL
CONDITION_COUNT=1
PATTERN[1]=Free space on volume
IS_CASE_SENSITIVE[1]=YES
IS_REGULAR_EXPRESSION[1]=NO
VALUE[1]=30
ARITHMETICAL_OPERATOR[1]=-gt
NEGATE_CONDITION_RESULT[1]=NO
============================================================
============= MAIN DECISION MAKING =============
COMPLEX condition evaluation
----------ATOMIC CONDITION 1------------
Searching for "Free space on volume" in JOBLOG of PATTERN_NUMBER_JOB
Pattern FOUND, performing further tests.
Searching for NUMBER withing row...
Number found=50. Evaluating arithmetical expression [ 50 -gt 30 ]
Arithmetical expression [ 50 -gt 30 ] evaluated as TRUE.
ATOMIC CONDITION RESULT [1]= TRUE
---------------------------------------
-------- COMPLEX CONDITION ----------
[ TRUE ]
CONDITION_RESULT=TRUE
TRUE: The result of complex condition is TRUE.
============================================================
============= Action on STOP Branch =============
Performing action CANCEL on job SYDNEY#0AAAAAAAAAAAAEET.B_DO_THE_BAD_THING
%cj SYDNEY#0AAAAAAAAAAAAEET.B_DO_THE_BAD_THING;schedid;noask
Command forwarded to batchman for SYDNEY#GBJ_PATTERN_NUM[(2113 12/17/07),
(0AAAAAAAAAAAAEET)].B_DO_THE_BAD_THING
============================================================
============= Action on RUN Branch =============
Performing action RELEASE on job SYDNEY#0AAAAAAAAAAAAEET.G_DO_THE_GOOD_THING
Releasing of job SYDNEY#0AAAAAAAAAAAAEET.G_DO_THE_GOOD_THING is NOT NECESSARY,
because priority=10
============================================================
============= Statistics of branch job BRANCH_1 =============
```

```
TRUE: The result of complex condition is TRUE.
For action CANCEL - RUN_BRANCH=G_DO_THE_GOOD_THING and
STOP_BRANCH=B_DO_THE_BAD_THING
BRANCH selected to STOP: B_DO_THE_BAD_THING
BRANCH selected to CONTINUE: G_DO_THE_GOOD_THING
CANCELED_JOBS: B_DO_THE_BAD_THING
PAUSED_JOB:
RELEASED_JOB:
============ END of branch job BRANCH_1 ==========
```

Parâmetros de Entrada Necessários

A Tabela 127 mostra os parâmetros necessários para o padrão no cenário de linha de padrão.

Tabela 127. Parâmetros de entrada para o cenário de comparação Numérica

Nome do parâmetro	Valor de parâmetro
CONDITION_SWITCH	COMPLEX
PATTERN_1	Free space on volume
VALUE_1	30
ARITHMETICAL_OPERATOR	-gt

É importante entender a ordem usada para transmitir os números para a expressão aritmética:

number_from_the_joblog compared_against *number_supplied_as_parameter*

`compared_against` é o operador aritmético especificado como parâmetro. Se você não especificar um operador, o valor padrão `-eq` (igual) será usado.

O exemplo a seguir mostra a definição de parâmetro. O texto é inserido no campo Comentários da definição de fluxo de tarefas.

```
BRANCH_1-BEGIN
CONDITION_SWITCH=COMPLEX
PATTERN_1=Free space on volume
VALUE_1=30
ARITHMETICAL_OPERATOR_1=-gt
BRANCH_1-END
```

Para obter uma descrição sobre como especificar os parâmetros para a tarefa de ramificação, consulte “Especificando os Parâmetros da Tarefa de Ramificação” na página 770.

Colocando a Tarefa de Ramificação no Fluxo de Tarefas

Coloque a tarefa de ramificação genérica no fluxo de tarefas após a tarefa-pai e renomeie o filho válido com o prefixo "G_" e o filho inválido com o prefixo "B_".

Além disso, siga a melhor prática e renomeie a tarefa de ramificação com um sufixo que consiste no caractere de sublinhado e em um valor numérico. Um nome típico para a primeira tarefa de ramificação em um fluxo de tarefas é BRANCH_1.

Cenário Complexo - Diversas Condições

Use o cenário de diversas condições para configurar diversas subcondições ao mesmo tempo.

Uso de Condição Complexa

Esta seção descreve como colocar todos os elementos atômicos de uma condição complexa juntos. A tarefa de ramificação genérica obtém o log da tarefa-pai e executa uma condição complexa nele:

- O log da tarefa *não deve* incluir o padrão `Error`.

e

- Uma das seguintes condições atômicas deve ser satisfeita:
  - O espaço livre no dispositivo Primário é conhecido e seu valor é maior que 50.
  - O espaço livre no dispositivo Secundário é conhecido e seu valor é maior que 60.

A Figura 54 mostra a definição para o cenário de ramificação complexa.

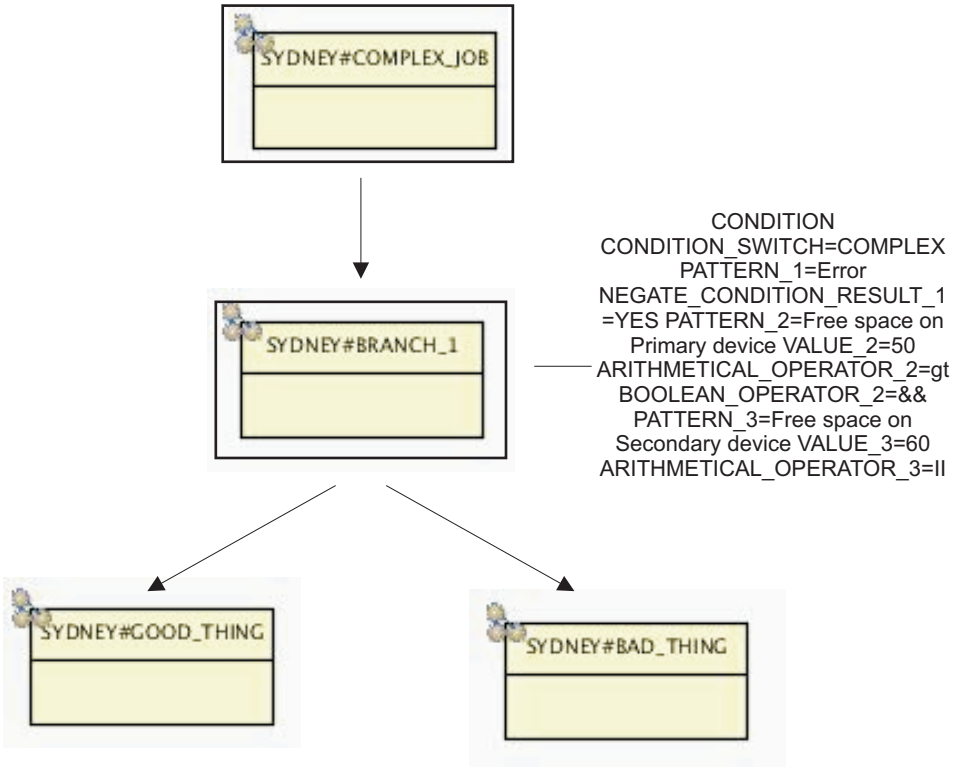


Figura 54. Definição de condição complexa

O log da tarefa mostra a saída da instância da tarefa de ramificação genérica:

```
============ START of branch job BRANCH_1 ==========
================ Job environment ===================
MASTER_PLATFORM=UNIX
STREAM_NAME=GBJ_COMPLEX
STREAM_CPU=SYDNEY
BRANCH_JOB_NAME=BRANCH_1
PARENT=COMPLEX_JOB
============================================================
============= Input parameters =============
CONDITION_SWITCH=COMPLEX
ACTION_SWITCH=CANCEL
CONDITION_COUNT=3
PATTERN[1]=Error
IS_CASE_SENSITIVE[1]=YES
IS_REGULAR_EXPRESSION[1]=NO
NEGATE_CONDITION_RESULT[1]=YES
PATTERN[2]=Free space on Primary device
IS_CASE_SENSITIVE[1]=YES
IS_REGULAR_EXPRESSION[1]=NO
VALUE[2]=50
ARITHMETICAL_OPERATOR[2]=-gt
BOOLEAN OPERATOR[2]=&&
NEGATE_CONDITION_RESULT[2]=NO
PATTERN[3]=Free space on Secondary device
IS_CASE_SENSITIVE[1]=YES
IS_REGULAR_EXPRESSION[1]=NO
VALUE[3]=60
ARITHMETICAL_OPERATOR[3]=-gt
BOOLEAN OPERATOR[3]=||
NEGATE_CONDITION_RESULT[3]=NO
============================================================
============= MAIN DECISION MAKING =============
COMPLEX condition evaluation
----------ATOMIC CONDITION 1------------
Searching for "Error" in JOBLOG of COMPLEX_JOB
Pattern NOT FOUND, condition evaluated as FALSE.
==NEGATED== ATOMIC CONDITION RESULT [1]= TRUE
```

```
----------ATOMIC CONDITION 2------------
Searching for "Free space on Primary device" in JOBLOG of COMPLEX_JOB
Pattern FOUND, performing further tests.
Searching for NUMBER withing row...
Number found=55. Evaluating arithmetical expression [ 55 -gt 50 ]
Arithmetical expression [ 55 -gt 50 ] evaluated as TRUE.
ATOMIC CONDITION RESULT [2]= TRUE
----------ATOMIC CONDITION 3------------
Searching for "Free space on Secondary device" in JOBLOG of COMPLEX_JOB
Pattern FOUND, performing further tests.
Searching for NUMBER withing row...
Number found=10. Evaluating arithmetical expression [ 10 -gt 60 ]
Arithmetical expression [ 10 -gt 60 ] evaluated as FALSE.
ATOMIC CONDITION RESULT [3]= FALSE
-------------------------------------
-------- COMPLEX CONDITION ----------
[ TRUE ] && [ TRUE ] || [ FALSE ]
CONDITION_RESULT=TRUE
TRUE: The result of complex condition is TRUE.
============================================================
============= Action on STOP Branch =============
Performing action CANCEL on job SYDNEY#0AAAAAAAAAAAAEEW.B_DO_THE_BAD_THING
%cj SYDNEY#0AAAAAAAAAAAAEEW.B_DO_THE_BAD_THING;schedid;noask
Command forwarded to batchman for SYDNEY#GBJ_COMPLEX[(2219 12/17/07)
(0AAAAAAAAAAAAEEW)].B_DO_THE_BAD_THING
============================================================
============= Action on RUN Branch =============
Performing action RELEASE on job SYDNEY#0AAAAAAAAAAAAEEW.G_DO_THE_GOOD_THING
Releasing of job SYDNEY#0AAAAAAAAAAAAEEW.G_DO_THE_GOOD_THING is NOT NECESSARY,
because priority=10
============================================================
============= Statistics of branch job BRANCH_1 =============
TRUE: The result of complex condition is TRUE.
For action CANCEL - RUN_BRANCH=G_DO_THE_GOOD_THING and
STOP_BRANCH=B_DO_THE_BAD_THING
BRANCH selected to STOP: B_DO_THE_BAD_THING
BRANCH selected to CONTINUE: G_DO_THE_GOOD_THING
CANCELED_JOBS: B_DO_THE_BAD_THING
PAUSED_JOB:
RELEASED_JOB:
============ END of branch job BRANCH_1 ==========
```

Parâmetros de Entrada Necessários

A Tabela 128 mostra os parâmetros necessários para o cenário de condição complexa.

Tabela 128. Parâmetros de entrada para o cenário de condição complexa

Nome do parâmetro	Valor de parâmetro
CONDITION_SWITCH	COMPLEX
PATTERN_1	Erro
NEGATE_CONDITION_RESULT_1	SIM
PATTERN_2	Free space on Primary device
VALUE_2	50
ARITHMETICAL_OPERATOR_2	-gt
BOOLEAN OPERATOR_2	&&
PATTERN_3	Free space on Secondary device
VALUE_3	60
ARITHMETICAL_OPERATOR_3	-gt
BOOLEAN OPERATOR_3	| |

O exemplo a seguir mostra a definição de parâmetro. O texto é inserido no campo Comentários da definição de fluxo de tarefas.

```
BRANCH_1-BEGIN
CONDITION_SWITCH=COMPLEX
PATTERN_1=Error
NEGATE_CONDITION_RESULT_1=YES
PATTERN_2=Free space on Primary device
VALUE_2=50
ARITHMETICAL_OPERATOR_2=-gt
BOOLEAN_OPERATOR_2=&&
PATTERN_3=Free space on Secondary device
VALUE_3=60
ARITHMETICAL_OPERATOR_3=-gt
BOOLEAN_OPERATOR_3=||
BRANCH_1-END
```

Para obter uma descrição sobre como especificar os parâmetros para a tarefa de ramificação, consulte “Especificando os Parâmetros da Tarefa de Ramificação” na página 770.

Colocando a Tarefa de Ramificação no Fluxo de Tarefas

Coloque a tarefa de ramificação genérica no fluxo de tarefas após a tarefa-pai e renomeie o filho válido com o prefixo "G_" e o filho inválido com o prefixo "B_".

Além disso, siga a melhor prática e renomeie a tarefa de ramificação com um sufixo que consiste no caractere de sublinhado e em um valor numérico. Um nome típico da primeira tarefa de ramificação em um fluxo de tarefas é BRANCH_1.

Parâmetros de Sequência Adicionais

A tarefa de ramificação genérica executa a procura de padrão usando o comando `grep`, que aceita vários parâmetros de entrada. Para refinar a procura de padrão para a tarefa de ramificação genérica, é possível usar `IS_CASE_SENSITIVE_`*`i`* e `IS_REGULAR_EXPRESSION_`*`i`*.

Use os parâmetros da seguinte forma:

IS_CASE_SENSITIVE_*i*
: Para ativar ou desativar um procura com distinção entre maiúsculas e minúsculas. O padrão é YES.

IS_REGULAR_EXPRESSION_*i*
: Para ativar ou desativar uma procura baseada em expressões regulares. O padrão é NO.

Nota: O sufixo_*i*, é o índice da subcondição específica.

Para obter uma descrição sobre como especificar os parâmetros para a tarefa de ramificação, consulte “Especificando os Parâmetros da Tarefa de Ramificação” na página 770.

Cenários Baseados no Tipo de Ação

Use uma tarefa de ramificação genérica baseada no tipo de ação para especificar a ação a ser executada na ramificação de execução e na ramificação de parada.

É possível concluir as seguintes ações:

- Parar ramificação
  - CANCEL: Toda a ramificação de parada é cancelada. Esta é a ação mais frequente na ramificação de parada.
  - PAUSE: A primeira tarefa da ramificação de parada está pausada (HOLD). Para obter informações mais detalhadas, consulte Cenário de Ações Pausar/Liberar
- Executar ramificação
  - Nenhuma ação: A ramificação de execução é executada. Esta é a ação mais frequente na ramificação de execução.
  - Liberar: Se a primeira tarefa da ramificação de execução estiver pausada (HOLD), aumente sua prioridade (RELEASE). Para obter informações mais detalhadas, consulte Cenário de Ações Pausar/Liberar
- Ação especial
  - SIGNAL: Esta ação não faz nada com nenhuma das ramificações. Ela recomenda qual confirmação o operador do Tivoli Workload Scheduler deve executar. Para obter informações mais detalhadas, consulte Cenário de Ação de Sinal.

Cenário de Ações Pausar e Liberar

Use essa tarefa de ramificação para gerenciar um fluxo de tarefas que seja sensível a alguns resultados de tarefa importantes, por exemplo, para considerar quando a ação executada por uma tarefa importante não foi concluída com sucesso.

uso de ações Pausar e Liberar

Use o cenário de pausa e liberação quando, mesmo que a tarefa de ramificação tenha identificado um estado de erro, você não deseja cancelar as ramificações. Em vez de cancelamento imediato, você faz a tarefa executar uma sequência de ações corretivas. Se as ações forem bem-sucedidas, a tarefa continuará como se o erro não tivesse ocorrido.

O fluxo do processo é semelhante ao seguinte exemplo:

1. Você possui uma tarefa importante que é seguida pela primeira tarefa de ramificação.
2. A tarefa de ramificação é seguida por uma ramificação válida e uma ramificação inválida: a ramificação válida (chamada OKbranch) inclui as tarefas a serem executadas se tudo for bem-sucedido e a ramificação inválida (chamada Correctivebranch) inclui uma sequência de tarefas para executar ações corretivas.
3. A primeira tarefa de ramificação avalia a condição executada na tarefa-pai (sua *tarefa importante*): se CONDITION=TRUE, tudo será bem-sucedido e a ramificação de parada será cancelada. Todas as tarefas na ramificação corretiva também serão canceladas, porque não é necessária nenhuma ação corretiva.

   Se CONDITION=FALSE, a ramificação de parada *não* será cancelada, mas será *pausada*, indicando que o filho válido (a primeira tarefa da ramificação de parada) será pausado. Ao pausar o filho válido, o OKbranch estará sendo RETIDO.
4. Enquanto OKbranch está sendo pausado, a ramificação corretiva inicia ações corretivas.
5. Após a conclusão da sequência de ações corretivas, a segunda tarefa de ramificação (colocada na ramificação corretiva) será enviada para avaliar o resultado das ações corretivas.

   Se as ações corretivas forem bem-sucedidas, o OKbranch será liberado; se as ações corretivas falharem, o OKbranch será cancelado e o fluxo de tarefas continuará executando a ramificação inválida da segunda tarefa de ramificação.

Geralmente, a ramificação inválida da segunda tarefa de ramificação contém apenas uma tarefa ABEND (uma tarefa que executa um comando *exit 1*). É uma boa prática encerrar a ramificação inválida da ramificação corretiva com a tarefa ABEND, porque isso assegura que o fluxo de tarefas será encerrado de forma anormal (as tarefas anteriores encerradas de forma anormal tinham Recovery Option=CONTINUE, portanto, elas não propagaram o status ABEND para o status do fluxo de tarefas final).

Nota: As duas tarefas de ramificação devem apontar para o mesmo filho válido. Isso é absolutamente crucial para que o processo funcione.

A Figura 55 na página 759 mostra a definição de fluxo de tarefas para o cenário de pausa/liberação. O parâmetro ACTION_SWITCH=PAUSE é definido para a primeira tarefa de ramificação. Não há parâmetros definidos para a segunda tarefa de ramificação.

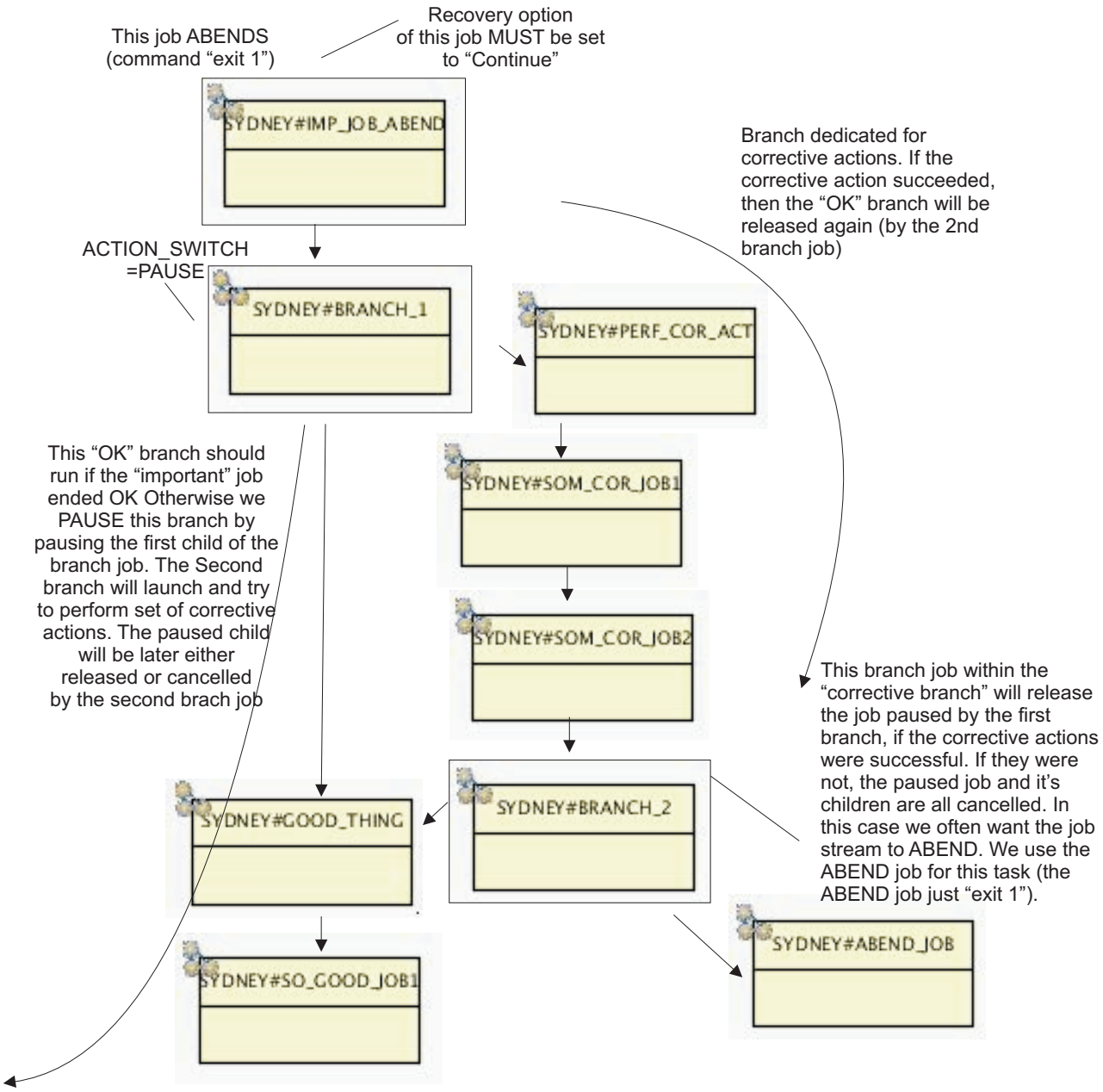


Figura 55. Definição de Ações Pausar e Liberar

O seguinte log da tarefa mostra a saída da primeira instância da tarefa de ramificação:

```
============ START of branch job BRANCH_1 ==========
================= Job environment ==================
MASTER_PLATFORM=UNIX
STREAM_NAME=GBJ_PAUSE
STREAM_CPU=SYDNEY
BRANCH_JOB_NAME=BRANCH_1
PARENT=IMPORTANT_JOB_ABEND
============================================================
============= Input parameters =============
CONDITION_SWITCH=PARENT_SUCCESS
ACTION_SWITCH=PAUSE
============================================================
============= MAIN DECISION MAKING =============
Evaluation dependent on PARENT_SUCCESS
FALSE: Searched for SUCC parent.
Status of PARENT JOB(IMPORTANT_JOB_ABEND) is ABEND.
============================================================
============= Action on STOP Branch =============
Performing action PAUSE on job SYDNEY#0AAAAAAAAAAAAEEX.G_DO_THE_GOOD_THING
%altpri SYDNEY#0AAAAAAAAAAAAEEX.G_DO_THE_GOOD_THING;schedid;0;noask
Command forwarded to batchman for SYDNEY#GBJ_PAUSE[(2300 12/17/07),
```

```
  (0AAAAAAAAAAAAEEX)].G_DO_THE_GOOD_THING
============================================================
============= Action on RUN Branch =============
Performing action RELEASE on job
SYDNEY#0AAAAAAAAAAAAEEX.B_PERFORM_CORRECTIVE_ACTIONS
Releasing of job SYDNEY#0AAAAAAAAAAAAEEX.B_PERFORM_CORRECTIVE_ACTIONS
is NOT NECESSARY,
because priority=10
============================================================
============= Statistics of branch job BRANCH_1 =============
FALSE: Searched for SUCC parent.
Status of PARENT JOB(IMPORTANT_JOB_ABEND) is ABEND.
For action PAUSE - RUN_BRANCH=B_PERFORM_CORRECTIVE_ACTIONS
and STOP_BRANCH=G_DO_THE_GOOD_THING
BRANCH selected to STOP: G_DO_THE_GOOD_THING
BRANCH selected to CONTINUE: B_PERFORM_CORRECTIVE_ACTIONS
CANCELED_JOBS:
PAUSED_JOB: G_DO_THE_GOOD_THING
RELEASED_JOB:
============ END of branch job BRANCH_1 ==========
```

O log da tarefa a seguir mostra a saída da segunda instância da tarefa de ramificação:

```
============ START of branch job BRANCH_2 ==========
============================================================
============= Job environment =============
MASTER_PLATFORM=UNIX
STREAM_NAME=GBJ_PAUSE
STREAM_CPU=SYDNEY
BRANCH_JOB_NAME=BRANCH_2
PARENT=SOME_CORRECTIVE_JOB_2
============================================================
============= Input parameters =============
CONDITION_SWITCH=PARENT_SUCCESS
ACTION_SWITCH=CANCEL
============================================================
============= MAIN DECISION MAKING =============
Evaluation dependent on PARENT_SUCCESS
TRUE: Searched for SUCC parent. Status of
PARENT JOB(SOME_CORRECTIVE_JOB_2) is SUCC.
============================================================
============= Action on STOP Branch =============
Performing action CANCEL on job SYDNEY#0AAAAAAAAAAAAEEX.B_ABEND_JOB
%cj SYDNEY#0AAAAAAAAAAAAEEX.B_ABEND_JOB;schedid;noask
Command forwarded to batchman for SYDNEY#GBJ_PAUSE[(2300 12/17/07),
   (0AAAAAAAAAAAAEEX)].B_ABEND_JOB
============================================================
============= Action on RUN Branch =============
Performing action RELEASE on job
SYDNEY#0AAAAAAAAAAAAEEX.G_DO_THE_GOOD_THING
Releasing SYDNEY#0AAAAAAAAAAAAEEX.G_DO_THE_GOOD_THING, because priority=0
%altpri SYDNEY#0AAAAAAAAAAAAEEX.G_DO_THE_GOOD_THING;schedid;10;noask
Command forwarded to batchman for SYDNEY#GBJ_PAUSE[(2300 12/17/07),
   (0AAAAAAAAAAAAEEX)].G_DO_THE_GOOD_THING
============================================================
============= Statistics of branch job BRANCH_2 =============
TRUE: Searched for SUCC parent. Status of
PARENT JOB(SOME_CORRECTIVE_JOB_2) is SUCC.
For action CANCEL - RUN_BRANCH=G_DO_THE_GOOD_THING and
STOP_BRANCH=B_ABEND_JOB
BRANCH selected to STOP: B_ABEND_JOB
BRANCH selected to CONTINUE: G_DO_THE_GOOD_THING
CANCELED_JOBS: B_ABEND_JOB
PAUSED_JOB:
RELEASED_JOB:G_DO_THE_GOOD_THING
============ END of branch job BRANCH_2 ==========
```

Parâmetros de Entrada Necessários

A Tabela 129 mostra os parâmetros necessários para a primeira tarefa de ramificação do cenário de pausa/liberação. A segunda tarefa de ramificação não requer parâmetros.

Tabela 129. Parâmetros de entrada para o cenário de pausa e liberação

Nome do parâmetro	Valor de parâmetro
ACTION_SWITCH	PAUSE

Nota:

- O parâmetro ACTION_SWITCH=PAUSE é necessário apenas para a primeira tarefa de ramificação no fluxo de tarefas. A segunda tarefa de ramificação deve ter ACTION SWITCH=CANCEL (este é o valor padrão).
- As duas tarefas de ramificação devem apontar para o mesmo filho válido.
- Cada tarefa de ramificação deve ter um sufixo diferente. Por exemplo, neste cenário simples de pausa e liberação, são usados dois nomes de tarefas de ramificação: BRANCH_1 e BRANCH_2.

A definição de parâmetro é semelhante ao seguinte exemplo. O texto é inserido no campo Comentários da definição de fluxo de tarefas.

```
BRANCH_1-BEGIN
ACTION_SWITCH=PAUSE
BRANCH_1-END
```

Para obter uma descrição sobre como especificar os parâmetros para a tarefa de ramificação, consulte “Especificando os Parâmetros da Tarefa de Ramificação” na página 770.

Colocando a Tarefa de Ramificação no Fluxo de Tarefas

Coloque a tarefa de ramificação genérica no fluxo de tarefas após a tarefa-pai e renomeie o filho válido com o prefixo "G_" e o filho inválido com o prefixo "B_".

A primeira tarefa de ramificação determina o OKbranch (seguido pelo filho válido) e o CorrectiveBranch (seguido pelo filho inválido). A tarefa que representa o filho válido deve ter o prefixo "G_", enquanto a tarefa que representa o filho inválido deve ter o prefixo "B_"."

As duas tarefas de ramificação devem apontar para o mesmo filho válido. Isso significa que o filho válido da primeira tarefa de ramificação deve ser idêntico ao filho válido da segunda tarefa de ramificação.

A melhor prática é colocar as tarefas ABEND como o filho inválido da segunda tarefa de ramificação. A tarefa ABEND apenas chama o comando do sistema *exit 1*, que causa o ENCERRAMENTO ANORMAL da tarefa.

Ter uma tarefa ABEND na ramificação inválida assegura que o status ABEND seja propagado também para o nível de fluxo de tarefas. As tarefas ENCERRADAS DE FORMA ANORMAL anteriormente não propagarão o estado ABEND para o nível de fluxo de tarefas se a tarefa tiver Recovery Option configurado como Continue. Para permitir a execução da tarefa de ramificação, você deve configurar Recovery Option como Continue para todos os pais da tarefa de ramificação.

Cenário de Diversas Ações Pausar e Liberar

Use o cenário de diversas ações pausar e liberar para executar uma sequência de ações corretivas e sair da ramificação corretiva quando alguma delas for bem-sucedida. Quando uma correção é concluída com êxito, a tarefa de ramificação cancela as ações corretivas restantes e libera o OKbranch.

Uso de diversas ações pausar e liberar

A Figura 56 na página 762 mostra a definição de fluxo de tarefas.

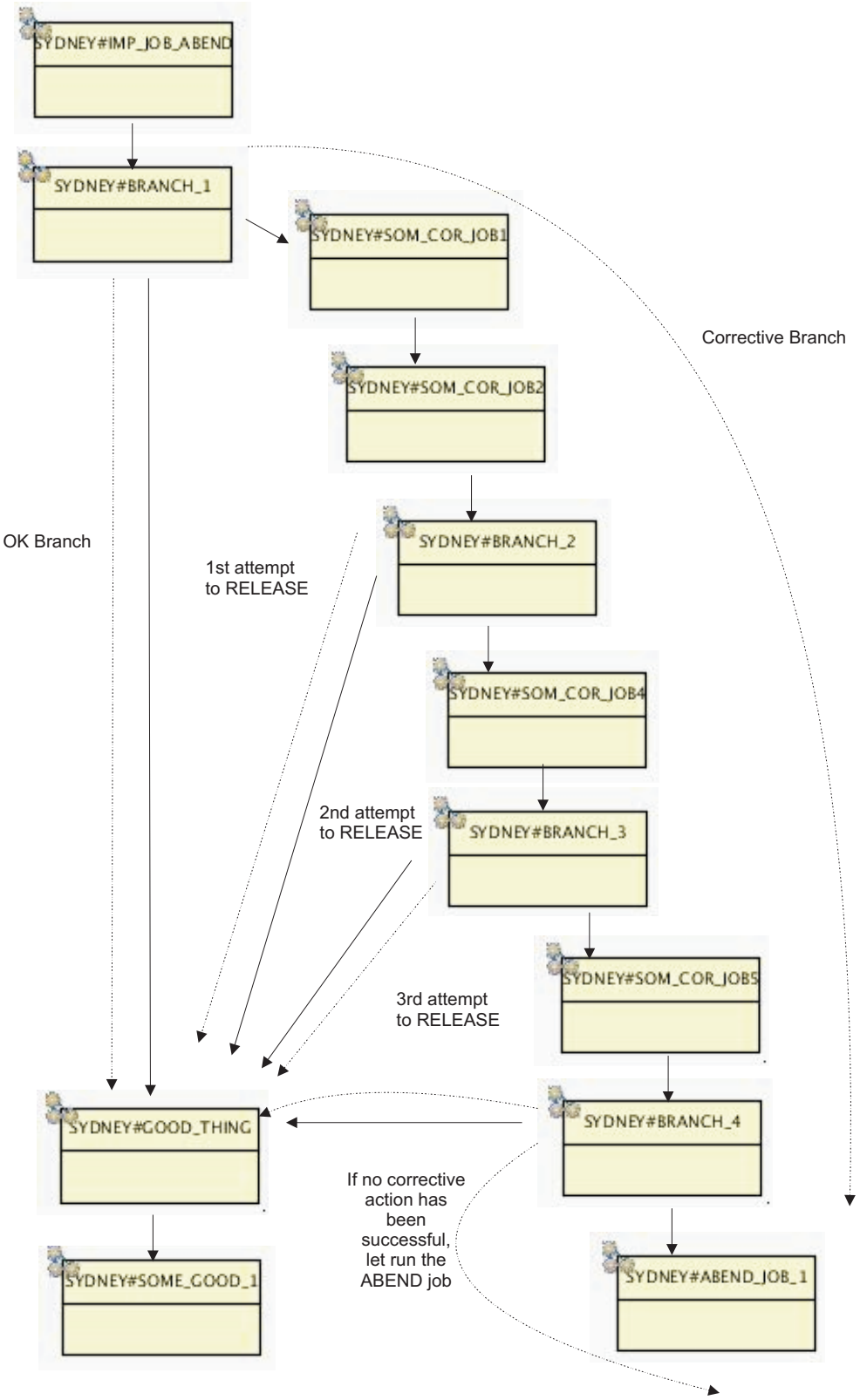


Figura 56. Definição de cenário de diversas ações pausar e liberar

Você deve configurar o parâmetro ACTION_SWITCH=PAUSE para *todas* as tarefas de ramificação, exceto a última. Portanto, neste cenário, as tarefas de ramificação

BRANCH_1, BRANCH_2 e BRANCH_3 devem ter ACTION_SWITCH=PAUSE. Se você não especificar este parâmetro, o filho válido será cancelado por uma tarefa de ramificação intermediária.

Todas as tarefas de ramificação devem apontar para o mesmo filho válido.

Certifique-se de que cada tarefa de ramificação tenha um sufixo diferente. Por exemplo, neste cenário, os nomes BRANCH_1, BRANCH_2, BRANCH_3 e BRANCH_4 são usados.

A melhor prática é configurar a tarefa ABEND como o filho inválido da última tarefa de ramificação. A tarefa ABEND chama o comando do sistema `exit 1`. Isso faz a tarefa ser ENCERRADA DE FORMA ANORMAL e seu status é propagado para o nível do fluxo de tarefas.

Cenário de Ação de Sinal

Use este cenário para ter uma tarefa de sinal que processa e armazena informações que são úteis para tomar uma decisão no log da tarefa.

Uso do cenário de ação de sinal

De uma perspectiva lógica, a tarefa de sinal e a tarefa de ramificação são diferentes em sua última etapa (a ação executada). Enquanto a tarefa de ramificação sempre cancela, pausa ou libera suas tarefas-filhas, a tarefa de sinal apenas registra uma recomendação para uma tomada de decisão. Em vez de bloquear o processo, o cenário de sinal seleciona a ramificação de execução e segue o fluxo de tarefas para continuar. Este cenário estende a abordagem já disponível com os prompts do Tivoli Workload Scheduler; ele representa a combinação de prompts com os recursos da tarefa de ramificação.

Neste cenário, há duas tarefas na seguinte ordem sequencial:

1. Sinal de tarefa
2. Tarefa de ramificação

Para o sinal de tarefa:

1. Especifique o parâmetro ACTION_SWITCH=SIGNAL e configure Recovery Option como CONTINUE. Na definição de fluxo de tarefas, configure o sinalizador `Requires Confirmation`.

   A tarefa de sinal executa a avaliação lógica, indicando que ela avalia a condição em relação às propriedades do pai, mas *não* cancela nem pausa nenhuma de suas tarefas-filhas.

   O sinalizador Requires Confirmation faz a tarefa de sinal parar o processamento do fluxo de tarefas. Após a conclusão da tarefa de sinal, ela permanece no status PEND. O log da tarefa de sinal mostra o processo de avaliação da condição concluído incluído na recomendação de confirmação. Isso significa que a tarefa de sinal avalia a condição e grava a recomendação (Confirmar SUCC ou Confirmar ABEND) em seu log da tarefa.
2. Consulte o log da tarefa de sinal e decida se confirmará a tarefa com um status SUCC ou ABEND.

   A seguinte tarefa de ramificação é iniciada apenas após a confirmação da tarefa de sinal. A tarefa de ramificação avalia o status configurado para a tarefa de sinal e determina a ramificação de execução e a ramificação de parada.

Esta última etapa representa a ramificação simples, que é descrita na "Cenário de Ramificação Simples" na página 733.

A Figura 57 mostra a definição de fluxo de tarefas para o cenário de sinal.

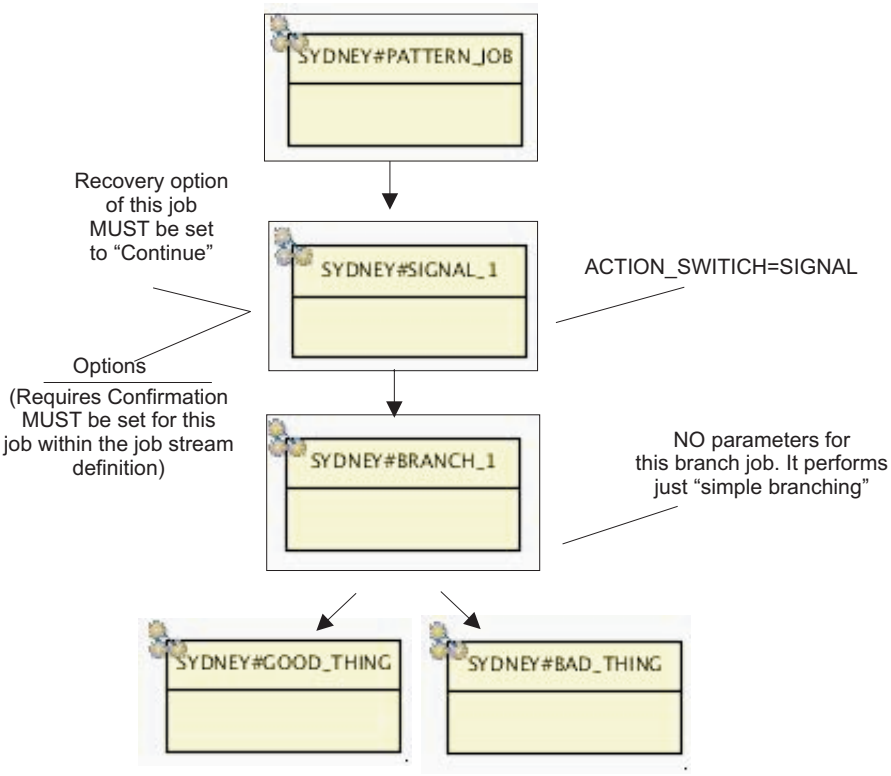


Figura 57. Definição de ação de sinal

O log da primeira tarefa de ramificação mostra que a tarefa está sendo RETIDA, porque está no status PEND. Este status requer sua confirmação, caso contrário, os sucessores da tarefa não serão executados.

```
============ START of branch job SIGNAL_1 ==========
================= Job environment ==================
MASTER_PLATFORM=UNIX
STREAM_NAME=GBJ_SIGNAL
STREAM_CPU=SYDNEY
BRANCH_JOB_NAME=SIGNAL_1
PARENT=PATTERN_JOB
============================================================
============= Input parameters =============
CONDITION_SWITCH=COMPLEX
ACTION_SWITCH=SIGNAL
CONDITION_COUNT=1
PATTERN[1]=completed successfully
IS_CASE_SENSITIVE[1]=YES
IS_REGULAR_EXPRESSION[1]=NO
NEGATE_CONDITION_RESULT[1]=NO
============================================================
============= MAIN DECISION MAKING =============
COMPLEX condition evaluation
----------ATOMIC CONDITION 1------------
Searching for "completed successfully" in JOBLOG of PATTERN_JOB
Pattern FOUND, performing further tests.
No additional value defined for specified pattern. Condition evaluated as TRUE.
ATOMIC CONDITION RESULT [1]= TRUE
---------------------------------------
-------- COMPLEX CONDITION ----------
[ TRUE ]
CONDITION_RESULT=TRUE
TRUE: The result of complex condition is TRUE.
============================================================
============= Statistics of branch job SIGNAL_1 =============
******Recommended confirmation for this job is SUCC.******
============ END of branch job SIGNAL_1 ==========
```

O log da segunda tarefa de ramificação mostra o processamento da seguinte instância da tarefa de ramificação.

```
============ START of branch job BRANCH_1 ==========
============================================================
============= Job environment =============
MASTER_PLATFORM=UNIX
STREAM_NAME=GBJ_SIGNAL
STREAM_CPU=SYDNEY
BRANCH_JOB_NAME=BRANCH_1
PARENT=SIGNAL_1
============================================================
============= Input parameters =============
CONDITION_SWITCH=PARENT_SUCCESS
ACTION_SWITCH=CANCEL
============================================================
============= MAIN DECISION MAKING =============
Evaluation dependent on PARENT_SUCCESS
TRUE: Searched for SUCC parent. Status of PARENT JOB(SIGNAL_1) is SUCC.
============================================================
============= Action on STOP Branch =============
Performing action CANCEL on job SYDNEY#0AAAAAAAAAAAAEI6.B_DO_THE_BAD_THING
%cj SYDNEY#0AAAAAAAAAAAAEI6.B_DO_THE_BAD_THING;schedid;noask
Command forwarded to batchman for SYDNEY#GBJ_SIGNAL[(0012 12/21/07),
   (0AAAAAAAAAAAAEI6)].B_DO_THE_BAD_THING
============================================================
============= Action on RUN Branch =============
Performing action RELEASE on job SYDNEY#0AAAAAAAAAAAAEI6.G_DO_THE_GOOD_THING
Releasing of job SYDNEY#0AAAAAAAAAAAAEI6.G_DO_THE_GOOD_THING is NOT NECESSARY,
   because priority=10
============================================================
============= Statistics of branch job BRANCH_1 =============
TRUE: Searched for SUCC parent. Status of PARENT JOB(SIGNAL_1) is SUCC.
For action CANCEL - RUN_BRANCH=G_DO_THE_GOOD_THING and
   STOP_BRANCH=B_DO_THE_BAD_THING
BRANCH selected to STOP: B_DO_THE_BAD_THING
BRANCH selected to CONTINUE: G_DO_THE_GOOD_THING
CANCELED_JOBS: B_DO_THE_BAD_THING
PAUSED_JOB:
RELEASED_JOB:
============ END of branch job BRANCH_1 ==========
```

Parâmetros de Entrada Necessários

A Tabela 130 mostra os parâmetros necessários para a tarefa de sinal. A seguinte tarefa de ramificação não usa nenhum parâmetro.

Tabela 130. Parâmetros de entrada para o cenário de ação de sinal

Nome do parâmetro	Valor de parâmetro
ACTION_SWITCH	SIGNAL

A definição de parâmetro é semelhante ao seguinte exemplo. O texto é inserido no campo Comentários da definição de fluxo de tarefas.

Nota: Esta definição também lista alguns parâmetros que são necessários *apenas* quando você usa a tarefa de sinal para uma procura de padrão no log da tarefa-pai, que é a principal vantagem do cenário de sinal. Use-o para executar a análise complexa de uma saída da tarefa, economizando tempo e tomando a decisão final de acordo com as informações coletadas.

```
SIGNAL_1-BEGIN
CONDITION_SWITCH=COMPLEX
PATTERN_1=completed successfully
ACTION_SWITCH=SIGNAL
SIGNAL_1-END
```

Para obter uma descrição sobre como especificar os parâmetros para a tarefa de ramificação, consulte “Especificando os Parâmetros da Tarefa de Ramificação” na página 770.

Colocando as Tarefas de Ramificação no Fluxo de Tarefas

Coloque a tarefa de sinal após a tarefa a ser avaliada. Designe à tarefa de sinal um nome que consista na sequência `SIGNAL` e um valor numérico (neste cenário, SIGNAL_1 é usado).

A seguinte tarefa de ramificação é o sucessor imediato da tarefa de sinal. Designe a ela um nome com o sufixo apropriado: se esta for a primeira tarefa de ramificação no fluxo de tarefas, o nome poderá ser BRANCH_1.

As tarefas-filhas da tarefa de ramificação devem ser nomeadas de acordo com as diretrizes comuns:

- O nome do filho válido deve começar com o prefixo "G_".
- O nome do filho inválido deve começar com o prefixo "B_".

Trabalhando com a Tarefa de Ramificação

É possível usar a tarefa de ramificação genérica em seu ambiente Tivoli Workload Scheduler.

Para entender como usar as tarefas de ramificação genérica, consulte:

- "Pré-requisitos para Executar Tarefas de Ramificação no Windows"
- "Definindo a Tarefa de Ramificação e a Tarefa de Sinal no Banco de Dados"
- "Colocando a Tarefa de Ramificação no Fluxo de Tarefas" na página 768
- "Usando a Tarefa ABEND" na página 769

Pré-requisitos para Executar Tarefas de Ramificação no Windows

Para executar a tarefa de ramificação genérica em sistemas operacionais Windows, certifique-se de que os seguintes pré-requisitos do sistema sejam atendidos.

- Como os sistemas operacionais Windows não podem interpretar nativamente shell scripts UNIX, você deve instalar um interpretador shell para usar o shell script *branch.sh* no gerenciador de domínio principal do Windows.
- O diretório `C:\cygwin\bin` deve apontar para o subdiretório bin do diretório de instalação do Cygwin. Se você tiver o Cygwin instalado em um diretório diferente do padrão, use o caminho correspondente.

Definindo a Tarefa de Ramificação e a Tarefa de Sinal no Banco de Dados

Para definir as tarefas de sinal de ramificação no banco de dados do Tivoli Workload Scheduler, é possível usar o Dynamic Workload Console ou o criador.

Definindo Duas Tarefas Apontando para um Shell Script

Para executar qualquer cenário de tarefa de ramificação, você deve definir uma tarefa de ramificação. Para executar um cenário de sinal, você também deve definir uma tarefa de sinal. As duas tarefas apontam para o mesmo arquivo de shell script com a diferença que:

- Para a tarefa de ramificação, você configura `Recovery Option` como STOP.
- Para a tarefa de sinal, você configura `Recovery Option` como CONTINUE.

Você tem duas tarefas diferentes que apontam para o mesmo arquivo de shell script para proteção contra o posicionamento incorreto da tarefa de ramificação ou da tarefa de sinal no fluxo de tarefas. A tarefa de ramificação verifica se está corretamente posicionada no fluxo de tarefas; se não estiver, ela será ENCERRADA DE FORMA ANORMAL. Para evitar a execução dos sucessores, a tarefa de ramificação tem `Recovery Option` configurado como STOP.

A tarefa de sinal executa a mesma verificação; se ela for posicionada incorretamente no fluxo de tarefas, a tarefa imprimirá uma mensagem de erro e será encerrada com um código de retorno diferente de zero. Mesmo que `Recovery`

`Option` esteja configurado como Continue, a tarefa não liberará seus sucessores da dependência, porque ela possui o sinalizador `Requires Confirmation` configurado na definição de fluxo de tarefas. Portanto, a tarefa de sinal permanece no status PEND aguardando a verificação de seu log da tarefa.

Para obter informações mais detalhadas, consulte "Cenário de Ação de Sinal" na página 763.

Definindo a Tarefa de Ramificação Usando o Dynamic Workload Console

Para definir uma tarefa de ramificação no banco de dados do Tivoli Workload Scheduler usando o Dynamic Workload Console, execute as seguintes etapas:

1. Abra o Designer de Carga de Trabalho.
2. Clique em **Novo** -> **Definição da Tarefa** -> **Nativo** e, de acordo com seu sistema operacional, **UNIX** ou **Windows**.
3. Na área de janela Geral, especifique os seguintes campos:
   - Nome: **BRANCH**
   - Estação de trabalho: seu gerenciador de domínio principal
   - Login: nome do usuário que envia e executa a tarefa no gerenciador de domínio principal
4. Na área de janela Tarefa:
   - Selecione o botão de opções **Script**.
   - De acordo com o sistema operacional:

     UNIX No campo Tarefa, especifique o caminho completo para o shell script `branch.sh`. Use barras (/).

     Windows
     No campo Tarefa, especifique dois caminhos:
     – O primeiro caminho aponta para Cygwin: forneça o caminho completo para o executável bash Cygwin. Use a notação padrão do Windows, ou seja, barras invertidas (\) como separadores de diretório.
     – O segundo caminho aponta para o shell script: forneça o caminho completo para o shell script `branch.sh`. Use barras (/). Se o caminho contiver espaços, coloque-o entre aspas duplas.

     Por exemplo, o valor para o campo Tarefa pode ser:

```
c:\cygwin\bin\bash\ "c:/Program Files/IBM/twa/scripts/branch.sh"
```

5. Na área de janela Opções de Recuperação, configure a Ação para **Parar**.

Nota: É possível usar os parâmetros do Tivoli Workload Scheduler como você usaria em qualquer outra definição de tarefa.

Definindo a Tarefa de Sinal Usando o Dynamic Workload Console

Para definir uma tarefa de sinal no banco de dados do Tivoli Workload Scheduler usando o Dynamic Workload Console, execute as seguintes etapas:

1. Abra o Designer de Carga de Trabalho.
2. Clique em **Novo** -> **Definição da Tarefa** -> **Nativo** e, de acordo com seu sistema operacional, **UNIX** ou **Windows**.
3. Na área de janela Geral, especifique os seguintes campos:

- Nome: **SIGNAL**
- Estação de trabalho: seu gerenciador de domínio principal
- Login: nome do usuário que envia e executa a tarefa no gerenciador de domínio principal

4. Na área de janela Tarefa:
   - Selecione o botão de opções **Script**.
   - De acordo com o sistema operacional:

     UNIX No campo Tarefa, especifique o caminho completo para o shell script `branch.sh`. Use barras (/).

     Windows
     No campo Tarefa, especifique dois caminhos:
     – O primeiro caminho aponta para Cygwin: forneça o caminho completo para o executável bash Cygwin. Use a notação padrão do Windows, ou seja, barras invertidas (\) como separadores de diretório.
     – O segundo caminho aponta para o shell script: forneça o caminho completo para o shell script `branch.sh`. Use barras (/). Se o caminho contiver espaços, coloque-o entre aspas duplas.

     Por exemplo, o valor para o campo Tarefa pode ser:

```
c:\cygwin\bin\bash\ "c:/Program Files/IBM/twa/scripts/branch.sh"
```

5. Na área de janela Opções de Recuperação, configure Ação como **Continuar**.

Definindo a Tarefa de Ramificação e a Tarefa de Sinal Usando o Composer

Para definir uma tarefa de ramificação e uma tarefa de sinal no banco de dados do Tivoli Workload Scheduler a partir do composer, execute as seguintes etapas:

1. Efetue logon no gerenciador de domínio principal como o usuário que tem as permissões para tarefas ADD.
2. De acordo com seu sistema operacional, crie uma nova definição de tarefa semelhante aos exemplos a seguir:

   UNIX

```
$JOBS
SYDNEY#BRANCH
SCRIPTNAME "/opt/ibm/tws_scripts/branch.sh"
STREAMLOGON tws
TASKTYPE UNIX
RECOVERY STOP

SYDNEY#SIGNAL
SCRIPTNAME "/opt/ibm/tws_scripts/branch.sh"
STREAMLOGON tws
TASKTYPE UNIX
RECOVERY CONTINUE
```

   Windows

```
$JOBS
HELSINKI#BRANCH
SCRIPTNAME "c:\cygwin\bin\bash \"c:/Program Files/IBM/tws/
                                        scripts/branch.sh\""
STREAMLOGON tws
TASKTYPE WINDOWS
RECOVERY STOP

$JOBS
HELSINKI#BRANCH
SCRIPTNAME "c:\cygwin\bin\bash \"c:/Program Files/IBM/tws/
                                        scripts/branch.sh\""
STREAMLOGON tws
TASKTYPE WINDOWS
RECOVERY CONTINUE
```

Colocando a Tarefa de Ramificação no Fluxo de Tarefas

É possível colocar a tarefa de ramificação genérica no fluxo de tarefas usando o Workload Dynamic Console.

Regras Gerais

Para colocar a tarefa de ramificação no fluxo de tarefas corretamente, certifique-se de que:

- Tenha apenas uma definição de tarefa de ramificação no banco de dados do Tivoli Workload Scheduler, conforme descrito em "Definindo a Tarefa de Ramificação e a Tarefa de Sinal no Banco de Dados" na página 766.
- Quando estiver inserindo a tarefa de ramificação no fluxo de tarefas, designe a ela um alias que descreva o posicionamento da tarefa de ramificação no fluxo de tarefas.
- Nomeie a primeira tarefa de ramificação BRANCH_1, a segunda BRANCH_2, etc.

Regras válidas para todos os cenários, exceto a ação de Sinal

As regras a seguir se aplicam a todos os cenários de uso da tarefa de ramificação, exceto para a ação SIGNAL:

- A tarefa de ramificação deve ter apenas um predecessor.
- A tarefa de ramificação deve ter exatamente duas tarefas-filhas, nomeadas da seguinte forma:
  - O nome do filho válido deve começar com `G_`.
  - O nome do filho inválido deve começar com `B_`.

Regras Válidas para o Cenário de Ação SIGNAL

As seguintes regras se aplicam à ação SIGNAL:

- Você deve usar duas tarefas sequencialmente ordenadas, da seguinte forma:
  - Sinal de tarefa
  - Tarefa de ramificação
- A tarefa de sinal deve ter apenas um filho, que representa a seguinte tarefa de ramificação.
- Para a tarefa de sinal, você deve configurar o parâmetro de entrada `ACTION_SWITCH=SIGNAL`.
- Para a tarefa de sinal, você deve configurar o sinalizador `Requires Confirmation` editando as propriedades da tarefa de ramificação.
- A tarefa de sinal deve ter apenas um predecessor.
- A tarefa de ramificação deve estar em conformidade com as regras descritas em "Regras válidas para todos os cenários, exceto a ação de Sinal ".

Usando a Tarefa ABEND

Use a tarefa ABEND quando desejar propagar o estado ABEND do status do fluxo de tarefas.

Se ocorrer algo errado em seu fluxo de tarefas e isso for detectado pela tarefa de ramificação, será possível propagar este *status inválido* no nível do fluxo de tarefas, colocando a tarefa ABEND na ramificação inválida que segue a tarefa de ramificação.

A tarefa ABEND emite apenas o comando `exit 1`, que faz a tarefa ser ENCERRADA DE FORMA ANORMAL, e assegura que o status ABEND seja propagado para o fluxo de tarefas. As tarefas ENCERRADAS DE FORMA ANORMAL anteriormente não propagam o status ABEND para o fluxo de tarefas,

porque elas têm a opção de recuperação configurada como Continue (isso é necessário para permitir a execução da tarefa de ramificação).

Para obter um exemplo de uma tarefa ABEND, consulte "Colocando a Tarefa de Ramificação no Fluxo de Tarefas" na página 768.

Especificando os Parâmetros da Tarefa de Ramificação

Especifique os parâmetros para a tarefa de ramificação no campo Comentários do fluxo de tarefas que contém a tarefa de ramificação afetada.

Nota: Certifique-se de especificar os parâmetros no campo Comentários, não o campo Descrição do editor de fluxo de tarefas.

Cada parâmetro deve ser colocado entre um *separador inicial* e um *separador final*, mesmo que você esteja especificando parâmetros apenas para uma tarefa de ramificação. Os separadores são construídos da seguinte forma:

Separador inicial
O nome da tarefa de ramificação, seguido pela sequência `-BEGIN`. Por exemplo, BRANCH_1-BEGIN.

Separador final
O nome da tarefa de ramificação, seguido pela sequência `-END`, por exemplo, BRANCH_1-END.

O exemplo a seguir mostra uma definição de parâmetro para uma ramificação de tarefa:

```
BRANCH_1-BEGIN
CONDITION_SWITCH=COMPLEX
PATTERN_1=Error
NEGATE_CONDITION_RESULT_1=YES
PATTERN_2=Free space on Primary device
VALUE_2=50
ARITHMETICAL_OPERATOR_2=-gt
BOOLEAN_OPERATOR_2=&&
PATTERN_3=Free space on Secondary device
VALUE_3=60
ARITHMETICAL_OPERATOR_3=-gt
BOOLEAN_OPERATOR_3=||
BRANCH_1-END
```

O exemplo a seguir mostra uma definição de parâmetro para um fluxo de tarefas com duas ramificações de tarefa:

```
BRANCH_1-BEGIN
CONDITION_SWITCH=COMPLEX
PATTERN_1=Error
NEGATE_CONDITION_RESULT_1=YES
PATTERN_2=Free space on Primary device
VALUE_2=50
ARITHMETICAL_OPERATOR_2=-gt
BOOLEAN_OPERATOR_2=&&
PATTERN_3=Free space on Secondary device
VALUE_3=60
ARITHMETICAL_OPERATOR_3=-gt
BOOLEAN_OPERATOR_3=||
BRANCH_1-END
BRANCH_2-BEGIN
CONDITION_SWITCH=COMPLEX
PATTERN_1=Backup of Primary device
VALUE_1=ERROR
NEGATE_CONDITION_RESULT_1=YES
BRANCH_2-END
```

O nome das tarefas de sinal deve começar com a sequência `SIGNAL`. Separadores de parâmetro também são necessários:

```
SIGNAL_1-BEGIN
ACTION_SWITCH=SIGNAL
SIGNAL_1-END
```

Referência de Parâmetros

A sintaxe e o significado dos parâmetros da tarefa de ramificação.

Tipos de parâmetro

Há dois tipos de parâmetros:

Fixo É possível especificar parâmetros fixos apenas uma vez na tarefa.

Indexado
É possível especificar parâmetros indexados várias vezes em uma tarefa, ou não especificá-los.

Parâmetros Fixos

É possível especificar os seguintes parâmetros fixos:

CONDITION_SWITCH
O tipo de condição executada na tarefa-pai. Ele pode ter os seguintes valores:

PARENT_SUCCESS
A condição é TRUE quando o pai é encerrado no estado SUCC. Este é o valor padrão. Para obter uma descrição desta condição, consulte "Cenários Baseados em Tipo de Condição" na página 733.

PARENT_ABEND
A condição é TRUE quando o pai é encerrado com o estado ABEND. Para obter uma descrição desta condição, consulte "Encerramento Anormal do Pai" na página 741.

COMPLEX
A condição é construída a partir de uma ou diversas subcondições conectadas a operadores booleanos (AND ou OR) e é avaliada com lógica booleana. Para obter uma descrição desta condição, consulte "Cenário Complexo - Diversas Condições" na página 754.

ACTION_SWITCH
A ação a ser executada na ramificação de parada. Ele pode ter os seguintes valores:

CANCEL
A ramificação de parada é cancelada. Este é o valor padrão.

PAUSE
A ramificação de parada é pausada. Para obter uma descrição desta condição, consulte "Cenário de Ações Pausar e Liberar" na página 758.

SIGNAL
Nenhuma ramificação é cancelada. Uma recomendação que requer sua confirmação está armazenada no log da tarefa. Para obter uma descrição desta condição, consulte "Cenário de Ação de Sinal" na página 763.

Parâmetros Indexados

Use parâmetros indexados apenas com o parâmetro CONDITION_SWITCH=COMPLEX especificado (caso contrário, eles serão ignorados durante o processo de ramificação).

Ao especificar CONDITION_SWITCH=COMPLEX, a tarefa de ramificação avalia uma condição complexa (uma condição com uma ou mais subcondições). Cada subcondição possui seu próprio*índice*, que começa com o número 1 e é incremental.

Para cada subcondição, você deve definir pelo menos um parâmetro Pattern_*i*, em que *i* representa o relacionamento entre a subcondição e o parâmetro correspondente. O sufixo é semelhante a *indexed_parameter_i*, em que *_i* é o sufixo. Por exemplo, os três parâmetros a seguir pertencem à mesma subcondição, que é a segunda na condição complexa; sua afinidade é expressa pelo sufixo_2, que representa o índice da subcondição:

```
PATTERN_2=find this row and number in this row
VALUE_2=50
ARITHMETICAL_OPERATOR_2=-lt
```

Agrupe os parâmetros relacionados a uma subcondição usando o mesmo índice. As subcondições são avaliadas separadamente e, em seguida, conectadas por operadores booleanos. A *condição complexa* é então avaliada.

Exemplos de Uso de Índice

O exemplo a seguir mostra os parâmetros a serem especificados se você desejar procurar três padrões no log da tarefa-pai. O índice age como o*contador incremental*. Os parâmetros pertencem a três subcondições separadas:

```
PATTERN_1=first pattern to find
PATTERN_2=second pattern to find
PATTERN_3=third pattern to find
```

Use a sintaxe a seguir para especificar os parâmetros relacionados à mesma subcondição (por exemplo, procurar um padrão, em seguida, um número na mesma linha e a comparação aritmética subsequente). Forneça o número e o operador aritmético como parâmetros de entrada.

```
PATTERN_1=Free space on Primary device
VALUE_1=50
ARITHMETICAL_OPERATOR_1=-gt
```

Significado de Parâmetros Indexados

A lista a seguir descreve os parâmetros indexados e seus valores possíveis. É possível usar todos esses parâmetros para criar uma subcondição única.

PATTERN_*i*

Procure um padrão de texto (por exemplo, `ended successfully`). Se o padrão for localizado, o resultado da condição será TRUE.

Para o parâmetro PATTERN_*i*, é possível especificar os seguintes parâmetros adicionais. Se PATTERN_*i* não for especificado, eles serão ignorados.

VALUE_*i*

Pode ser STRING ou NUMERIC. Isso é determinado automaticamente ao ler o parâmetro de valor específico durante a inicialização da tarefa de ramificação.

O valor especificado por VALUE_*i* é procurado*na mesma linha* identificada pela procura da sequência indicada por PATTERN_*i*.

Dois tipos de valores são possíveis e eles são determinados automaticamente analisando o conteúdo do parâmetro VALUE_*i*:

Valor de sequência

Procura outro padrão de texto na mesma linha.

Se os dois padrões forem localizados na mesma linha, o resultado da condição será TRUE.

Para obter uma descrição desta função, consulte "Ramificação Complexa - Padrão na Linha de Padrão" na página 747.

Valor numérico
Procura o valor numérico na mesma linha.

O operador aritmético especificado é então usado para executar a comparação aritmética. Se a comparação aritmética for bem-sucedida, a condição será TRUE. Um operador aritmético específico é definido para cada valor numérico.

Para obter uma descrição desta função, consulte "Ramificação Complexa - Comparação de Valor Numérico" na página 751.

ARITHMETICAL_OPERATOR_*i*
O operador usado para a comparação aritmética.

NEGATE_CONDITION_RESULT_*i*
Este argumento nega o resultado da subcondição específica, significando que ele troca o resultado TRUE ou FALSE da subcondição.

BOOLEAN_OPERATOR_*i*
As subcondições definidas são associadas pelo operador booleano AND ou OR. É possível usar o operador booleano porque `i=2`. Isso significa que o índice do operador booleano deve ser pelo menos 2.

Por exemplo, você possui dois parâmetros na lista. Cada um deles representa uma subcondição. Cada subcondição é avaliada separadamente e seu resultado é retornado como TRUE ou FALSE. Para avaliar a condição inteira, você deve associar os resultados específicos.

O significado de BOOLEAN_OPERATOR_*i* é que o resultado da conexão *desta* subcondição com o resultado da subcondição *precedente* usa o operador booleano AND ou OR.

Tabelas de referência

A Tabela 131 descreve os parâmetros, seus valores possíveis e padrão.

Tabela 131. Parâmetros e Valores

Nome do parâmetro	Valores Possíveis	Valor padrão
CONDITION_SWITCH	PARENT_SUCCESS PARENT_ABEND COMPLEX	PARENT_SUCCESS
ACTION_SWITCH	CANCEL PAUSE SIGNAL	CANCEL (para tarefas de ramificação) SIGNAL (para tarefas de sinal)
PATTERN_*i*, em que *i* é o índice incremental	Qualquer sequência	
VALUE_*i*, em que *i* é o índice incremental	Qualquer sequência Qualquer valor numérico (número inteiro ou real)	

Tabela 131. Parâmetros e Valores (continuação)

Nome do parâmetro	Valores Possíveis	Valor padrão
ARITHMETICAL_OPERATOR_*i*, em que *i* é o índice incremental	-lt -le -eq -ne -ge -gt	-eq
IS_CASE_SENSITIVE_*i*, em que *i* é o índice incremental	SIM NÃO	SIM
IS_REGULAR_EXPRESSION_*i*, em que *i* é o índice incremental	SIM NÃO	NÃO
NEGATE_CONDITION_RESULT_*i*, em que *i* é o índice incremental	SIM NÃO	NÃO
BOOLEAN OPERATOR_*i*, em que *i* é o índice incremental	&& ||	&&

Os valores para operadores aritméticos e booleanos usam a sintaxe do UNIX. Seus significados são mostrados na Tabela 132.

Tabela 132. Descrição de operadores aritméticos

Valor de parâmetro UNIX	Interpretação de valor de parâmetro	Significado do valor de parâmetro
-lt	<	Menor que
-le	<=	Menor ou igual a
-eq	=	Igual a
-ne	!=	Diferente de
-ge	>=	Maior ou igual a
-gt	>	Maior que
&&	AND	AND Lógico
||	OR	OR Lógico

Distinção entre Maiúsculas e Minúsculas

As regras a seguir se aplicam ao especificar um parâmetro:

- Os nomes de parâmetros não fazem distinção entre maiúsculas e minúsculas.
- Os valores de parâmetros fixos não fazem distinção entre maiúsculas e minúsculas.
- Os padrões a serem procurados no log da tarefa-pai fazem distinção entre maiúsculas e minúsculas. Para substituir este comportamento, especifique IS_CASE_SENSITIVE_*i*=NO, em que *i* representa o índice de subcondição atual. Por exemplo, para o seguinte parâmetro:

  ```
  PATTERN_2=some text
  ```

  alterne a procura de padrão de distinção entre maiúsculas e minúsculas especificando este parâmetro adicional:

  ```
  IS_CASE_SENSITIVE_2=NO
  ```

  Se estiver procurando dois padrões na mesma linha, este parâmetro afetará ambos.
- Os separadores de parâmetros necessários para definir a tarefa de ramificação fazem distinção entre maiúsculas e minúsculas. Você deve especificar os separadores em letras maiúsculas, caso contrário, eles não serão lidos e os valores padrão serão usados.

Exemplos de Condições de Amostra

As seções a seguir resumem como construir conjuntos de parâmetros simples ou mais complexos.

Ramificação Simples, Cenários de Ramificação Longa

Esta função usa os valores padrão:

- CONDITION_SWITCH=PARENT_SUCCESS
- ACTION_SWITCH=CANCEL

Não é necessário especificar nenhum parâmetro de entrada para estes cenários. Os valores padrão são fornecidos automaticamente pela tarefa de ramificação.

Pause action

Essa função precisa de pelo menos um parâmetro fixo. Você deve usar pelo menos duas tarefas de ramificação para implementar o cenário de pausa e de liberação.

Para usar a função de pausa e liberação, você deve substituir o comportamento padrão para a primeira tarefa de ramificação. O nome da tarefa de ramificação neste exemplo é BRANCH_1. Se o nome de sua tarefa de ramificação tiver um sufixo diferente (por exemplo, BRANCH_5), você deverá ajustar os nomes dos separadores.

A segunda tarefa de ramificação na abordagem de pausa e liberação não precisa de nenhum parâmetro de entrada, se for dependente apenas do status de seu pai (SUCC).

A seguir está a sintaxe completa juntamente com os separadores de parâmetros:

```
BRANCH_1-BEGIN
ACTION_SWITCH=PAUSE
BRANCH_1-END
```

Não são fornecidos parâmetros indexados. Os parâmetros indexados podem ser usados apenas em combinação com o parâmetro fixo CONDITION_SWITCH=COMPLEX. Como o parâmetro CONDITON_SWITCH não está incluído, o valor padrão CONDITION_SWITCH=PARENT_SUCCESS foi usado.

Para obter informações sobre a combinação da ação pausar juntamente com a condição complexa, consulte “Condição Complexa com Ação Pausar” na página 778.

Procura de Padrão Único

Essa função é representada por uma subcondição e precisa de um parâmetro fixo e de um indexado.

O exemplo a seguir mostra a sintaxe completa juntamente com os separadores de parâmetros. O nome da tarefa de ramificação é BRANCH_1. Se o nome de sua tarefa de ramificação tiver um sufixo diferente (por exemplo, BRANCH_5), você deverá ajustar os nomes dos separadores.

```
BRANCH_1-BEGIN
CONDITION_SWITCH=COMPLEX
PATTERN_1=find this text
BRANCH_1-END
```

Há apenas um parâmetro indexado representando a única subcondição.

Procura de Padrão Negada

Essa função é representada por uma subcondição e requer um parâmetro fixo e dois indexados.

O exemplo a seguir mostra a sintaxe completa juntamente com os separadores de parâmetros. O nome da tarefa de ramificação é BRANCH_1. Se o nome de sua tarefa de ramificação tiver um sufixo diferente (por exemplo, BRANCH_5), você deverá ajustar os nomes dos separadores.

```
BRANCH_1-BEGIN
CONDITION_SWITCH=COMPLEX
PATTERN_1=do not find this text
NEGATE_CONDITION_RESULT_1=YES
BRANCH_1-END
```

Os dois parâmetros indexados possuem o mesmo sufixo, porque eles pertencem à mesma subcondição.

Procura de Diversos Padrões

Essa função é representada por várias subcondições e precisa de um parâmetro fixo e vários indexados.

Neste exemplo, você procura dois padrões independentes no log da tarefa-pai. Você cria duas subcondições separadas, cada uma com seu próprio índice (1 e 2). Você também deve especificar se os *dois* padrões devem ser localizados (usando o operador Booleano*AND*) ou se localizar *pelo menos um* padrão é suficiente (operador Booleano*OR*).

O exemplo a seguir mostra a sintaxe completa, juntamente com separadores de parâmetros. O nome da tarefa de ramificação é BRANCH_1. Se o nome de sua tarefa de ramificação tiver um sufixo diferente (por exemplo, BRANCH_5), você deverá ajustar os nomes dos separadores.

```
BRANCH_1-BEGIN
CONDITION_SWITCH=COMPLEX
PATTERN_1=find this text
PATTERN_2=find also this text
BOOLEAN_OPERATOR_2=&&
BRANCH_1-END
```

Os parâmetros indexados possuem sufixos diferentes.

O parâmetro PATTERN_1 pertence à primeira subcondição e o parâmetro PATTERN_2 pertence à segunda subcondição. O parâmetro BOOLEAN_OPERATOR_2 especifica como a segunda subcondição é associada à primeira subcondição.

Padrão na Procura de Padrão

Essa função é representada por uma subcondição e precisa de um parâmetro fixo e dois indexados.

Neste exemplo, você procura o padrão de texto no log da tarefa-pai. Em seguida, procura outro padrão na mesma linha. Todos os parâmetros indexados pertencem a uma subcondição.

O exemplo a seguir mostra a sintaxe completa, juntamente com separadores de parâmetros. O nome da tarefa de ramificação é BRANCH_1. Se o nome de sua tarefa de ramificação tiver um sufixo diferente (por exemplo, BRANCH_5), você deverá ajustar os nomes dos separadores.

```
BRANCH_1-BEGIN
CONDITION_SWITCH=COMPLEX
PATTERN_1=find this text
VALUE_1=and also this text on the same row
BRANCH_1-END
```

Os dois parâmetros indexados possuem o mesmo sufixo, porque eles pertencem à mesma subcondição.

Procura de Padrão com Comparação Numérica

Essa função é representada por uma subcondição e requer um parâmetro fixo e três indexados.

Você procura um padrão de texto no log da tarefa-pai. Em seguida, procura um número na mesma linha. Você deseja comparar este número com outro número fornecido como o parâmetro de entrada. Para a comparação aritmética, use o operador aritmético fornecido como o parâmetro de entrada.

Todos os parâmetros indexados pertencem a uma subcondição.

O exemplo a seguir mostra a sintaxe completa, juntamente com separadores de parâmetros. O nome da tarefa de ramificação é BRANCH_1. Se o nome de sua tarefa de ramificação tiver um sufixo diferente (por exemplo, BRANCH_5), você deverá ajustar os nomes dos separadores.

```
BRANCH_1-BEGIN
CONDITION_SWITCH=COMPLEX
PATTERN_1=Total backup size
VALUE_1=500
ARITHMETICAL_OPERATOR_1=-lt
BRANCH_1-END
```

Todos os três parâmetros indexados possuem o mesmo sufixo, porque eles pertencem à mesma subcondição.

O significado das funções chamadas pelos parâmetros especificados pode ser representado como a seguir:

1. Obter o log da tarefa-pai.
2. Extraia a linha que contém a sequência `Total backup size`.
3. If (linha localizada):
4. Continuar com a próxima etapa
5. Else return FALSE
6. Tentar extrair o valor numérico da linha.
7. If (número localizado):
8. Continuar com a próxima etapa
9. Else return FALSE
10. If (number_from_joblog < 50):
11. Then return TRUE
12. Else return FALSE

Combinação de Ações Procurar e Pausar do Padrão

Uma condição não padrão pode ser combinada com uma ação não padrão. Isso significa que usamos entradas para CONDITION_SWITCH e ACTION_SWITCH neste cenário.

Demonstramos como combinar procura de padrão único com a ação pausar.

Essa função é representada por uma subcondição e precisa de dois parâmetros fixos e de um parâmetro indexado.

O exemplo a seguir mostra a sintaxe completa, juntamente com separadores de parâmetros. O nome da tarefa de ramificação é BRANCH_1. Se o nome de sua tarefa de ramificação tiver um sufixo diferente (por exemplo, BRANCH_5), você deverá ajustar os nomes dos separadores.

```
BRANCH_1-BEGIN
CONDITION_SWITCH=COMPLEX
ACTION_SWITCH=PAUSE
PATTERN_1=find this text
BRANCH_1-END
```

Os dois parâmetros fixos são substituídos. O parâmetro indexado um pertence à subcondição um.

Ação de Sinal

São necessários separadores diferentes para o cenário de sinal.

Duas tarefas estão gerenciando o fluxo de trabalho do fluxo de tarefas:

- A tarefa de sinal
- A tarefa de ramificação

Apenas a tarefa de sinal usa os parâmetros de entrada.

O nome da tarefa de sinal consiste na sequência `SIGNAL` e no sufixo, portanto, os separadores de parâmetros parecem *diferentes*. Além disso, para a tarefa de sinal, as regras comuns são válidas. Os separadores de parâmetros devem parecer exatamente como o nome da tarefa relacionada.

Este cenário mostra a diferença juntamente com os parâmetros para a ação de sinal. A ação de sinal é combinada com a condição baseada na procura de padrão no log da tarefa-pai.

A função de sinal é representada por um parâmetro fixo. A procura de padrão requer um parâmetro fixo para o tipo de condição e um parâmetro indexado que especifica a procura de padrão.

O exemplo a seguir mostra a sintaxe completa, juntamente com separadores de parâmetros. O nome da tarefa de ramificação é BRANCH_1. Se o nome de sua tarefa de ramificação tiver um sufixo diferente (por exemplo, BRANCH_5), você deverá ajustar os nomes dos separadores.

```
SIGNAL_1-BEGIN
CONDITION_SWITCH=COMPLEX
ACTION_SWITCH=SIGNAL
PATTERN_1=find this text
SIGNAL_1-END
```

Condição Complexa com Ação Pausar

A condição complexa pode ser combinada com a ação não padrão.

A condição complexa é descrita em "Cenário Complexo - Diversas Condições" na página 754. Neste cenário, a condição complexa é combinada com a ação pausar.

O exemplo a seguir mostra a sintaxe completa, juntamente com separadores de parâmetros. O nome da tarefa de ramificação é BRANCH_1. Se o nome de sua tarefa de ramificação tiver um sufixo diferente (por exemplo, BRANCH_5), você deverá ajustar os nomes dos separadores.

```
BRANCH_1-BEGIN
CONDITION_SWITCH=COMPLEX
ACTION_SWITCH=PAUSE
PATTERN_1=Error
NEGATE_CONDITION_RESULT_1=YES
PATTERN_2=Free space on Primary device
VALUE_2=50
ARITHMETICAL_OPERATOR_2=-gt
BOOLEAN_OPERATOR_2=&&
PATTERN_3=Free space on Secondary device
VALUE_3=60
ARITHMETICAL_OPERATOR_3=-gt
BOOLEAN_OPERATOR_3=||
BRANCH_1-END
```

O significado da definição é o seguinte:

- Subcondition_1: Se você localizar o padrão `Error`, return FALSE, else return TRUE. O resultado invertido é realizado pelo parâmetro NEGATE_CONDITION_RESULT_1.
- Subcondition_2: Procure a linha que contém `Free space on Primary device`. Extraia o número da linha. Se o número for maior que 50, return TRUE, else return FALSE.
- Subcondition_3: Procure a linha que contém `Free space on Secondary device`. Extraia o número dessa linha. Se o número for maior que 60, return TRUE, else return FALSE.
- Associe essas três subcondições usando operadores Booleanos para que a condição completa seja construída da seguinte forma:
  1. If (subcondition_1 = TRUE) AND (subcondition_2=TRUE) OR (subcondition_3=FALSE)
  2. Then return TRUE
  3. Else return FALSE
  4. If (complex_condition_result=TRUE)
  5. Then CANCEL bad_branch
  6. Else PAUSE the good_branch

Para obter informações adicionais sobre os conceitos de pausa e liberação, consulte “Cenário de Ações Pausar e Liberar” na página 758.

Diversas Tarefas de Ramificação em um Fluxo de Tarefas

É possível definir parâmetros para várias tarefas de ramificação que estão definidas no mesmo fluxo de tarefas. Ter mais de uma tarefa de ramificação em um fluxo de tarefas significa que os diferentes conjuntos de parâmetros são colocados entre *separadores* diferentes.

O exemplo a seguir mostra a sintaxe completa, juntamente com separadores de parâmetros. Os nomes das tarefas de ramificação são BRANCH_1 e BRANCH_2. Se o nome de suas tarefas de ramificação tiverem sufixos diferentes (por exemplo, BRANCH_5 e BRANCH_6), você deverá ajustar os nomes dos separadores.

```
BRANCH_1-BEGIN
CONDITION_SWITCH=COMPLEX
PATTERN_1=Error
NEGATE_CONDITION_RESULT_1=YES
PATTERN_2=Free space on Primary device
VALUE_2=50
ARITHMETICAL_OPERATOR_2=-gt
BOOLEAN_OPERATOR_2=&&
PATTERN_3=Free space on Secondary device
VALUE_3=60
ARITHMETICAL_OPERATOR_3=-gt
BOOLEAN_OPERATOR_3=||
```

```
BRANCH_1-END

BRANCH_2-BEGIN
CONDITION_SWITCH=COMPLEX
PATTERN_1=Backup of Primary device
VALUE_1=ERROR
NEGATE_CONDITION_RESULT_1=YES
BRANCH_2-END
```

Notas Importantes sobre a Tarefa de Ramificação

Esta seção destaca algumas considerações e suposições para o design de tarefa de ramificação genérico.

- O nome da tarefa de ramificação no banco de dados é BRANCH. A tarefa deve ter configurado a propriedade Recovery options=STOP.
- O nome da tarefa de sinal no banco de dados é SIGNAL. A tarefa deve ter configurado a propriedade Recovery options=CONTINUE.
- O nome da tarefa de ramificação colocado no fluxo de tarefas deve consistir no nome da tarefa de ramificação (BRANCH) e no *sufixo*. O sufixo reflete a posição da tarefa de ramificação no fluxo de tarefas. Por exemplo, BRANCH_3 é a terceira tarefa de ramificação no fluxo de tarefas.
- O nome da tarefa de sinal colocada no fluxo de tarefas deve consistir no nome da tarefa de sinal (SIGNAL) e no sufixo. O sufixo reflete a posição da tarefa de sinal no fluxo de tarefas. Por exemplo, SIGNAL_3 é a terceira tarefa de sinal no fluxo de tarefas.

  O contador de sufixos para tarefas de sinal é diferente do contador de sufixos de tarefas de ramificação, portanto, as tarefas BRANCH_3 e SIGNAL_3 podem existir em um fluxo de tarefas ao mesmo tempo, mesmo que seu sufixo seja idêntico.
- A tarefa de ramificação possui apenas um pai. Isso significa que a tarefa de ramificação deve ter a dependência FOLLOWS configurada exatamente em uma tarefa no mesmo fluxo de tarefas. Isso é verificado durante a inicialização da tarefa de ramificação. Se esta condição não for atendida, a tarefa de ramificação será ENCERRADA DE FORMA ANORMAL.

  A conexão da tarefa de ramificação com mais de um pai é interrompida pela tarefa de ramificação e causa o ENCERRAMENTO DE FORMA ANORMAL da tarefa de ramificação.
- Para o pai que é avaliado com relação ao status de resultado (SUCC ou ABEND), você deve configurar a seguinte propriedade na definição de tarefa, não na definição de fluxo de tarefas:

  ```
  Recovery options=Continue
  ```
- Para os seguintes cenários de uso da tarefa de ramificação, a tarefa de ramificação deve ter exatamente duas tarefas-filhas.
  - ACTION_SWITCH=CANCEL (valor padrão)
  - ACTION_SWITCH=PAUSE
  - As tarefas-filhas devem ser identificadas como o filho válido e o filho inválido. As tarefas-filhas devem ser nomeadas da seguinte forma:
    - A tarefa que representa o filho válido deve ter um nome que começa com "G_". Não é necessário renomear a *definição de tarefa*, renomeie o*alias da tarefa* no fluxo de tarefas.

      Por exemplo, o nome da tarefa no banco de dados é START_BACKUP. Para usar esta tarefa como o filho válido da tarefa de ramificação, coloque esta tarefa no fluxo de tarefas e, no fluxo de tarefas, designe à tarefa o alias G_START_BACKUP.

- A tarefa que representa o filho inválido deve ter um nome que começa com "B_". Não é necessário renomear a definição de tarefa, renomeie o alias da tarefa no fluxo de tarefas.

  Por exemplo, o nome da tarefa no banco de dados é RESUME_DATABASE. Para usar esta tarefa como o filho inválido da tarefa de ramificação, coloque esta tarefa no fluxo de tarefas e, no fluxo de tarefas, designe à tarefa o alias B_RESUME_DATABASE.

  A contagem das tarefas-filhas e seus prefixos corretos são verificados durante a inicialização da tarefa de ramificação. Se estas condições não forem atendidas, a tarefa de ramificação será ENCERRADA DE FORMA ANORMAL.

- O uso de caracteres de aspas não está implementado na versão atual da tarefa de ramificação. Se você especificar caracteres aspas simples nos parâmetros PATTERN_*i* ou VALUE_*i*, os caracteres aspas simples serão removidos automaticamente.
- Ao avaliar a condição complexa, o parâmetro de procura inicial é um padrão de texto. Neste padrão de texto, é procurado o log da tarefa-pai. Se houver mais de uma linha que inclui o padrão procurado, apenas a primeira linha correspondida será identificada. As outras linhas correspondidas serão ignoradas.

  O conteúdo da primeira linha identificada é avaliado por outra subcondição, conforme mostrado no padrão em cenários de comparação de linha de padrão e de valor numérico.
- Ao extrair o valor numérico da linha de padrão, pode haver apenas um valor numérico na linha. Os valores numéricos de número inteiro e de tipo real são aceitos.

  Mais de um número na linha de padrão identificada resultará na extração do valor numérico incorreto; todos os números da linha serão associados e não representarão um valor significativo.
- Ao usar o cenário de sinal, a tarefa de sinal deve ter exatamente um filho. O filho deve ter uma tarefa de ramificação subsequente e seu nome deve estar de acordo com as convenções de nomenclatura para a tarefa de ramificação genérica.
- Ao usar o par de ações PAUSE e RELEASE, o processo é semelhante ao seguinte:
  – A primeira tarefa de ramificação configura a prioridade da tarefa que está prestes a ser pausada em 0.
  – A segunda tarefa de ramificação configura a prioridade da tarefa que está prestes a ser liberada em 10.

  Essa função é codificada permanentemente. Não há nenhum mecanismo implementado que permitiria "lembrar" a prioridade da tarefa original.
- Todas as informações sobre o fluxo de tarefas, no qual a tarefa de ramificação é executada, são extraídas do *banco de dados* usando o comando *composer* quando a tarefa de ramificação é iniciada. Isso significa que a tarefa de ramificação não reflete nenhuma instância da tarefa que foi *enviada para* o fluxo de tarefas, mas não existe na definição de fluxo de tarefas.

Apêndice E. Acessibilidade

Os recursos de acessibilidade ajudam usuários com deficiências físicas, como mobilidade restrita ou visão limitada, a usar os produtos de software com êxito. Os principais recursos de acessibilidade neste produto permitem que os usuários façam o seguinte:

- Utilizem tecnologias assistidas, como software de leitor de tela e sintetizador de voz digital, para ouvir os itens que são exibidos na tela. Consulte a documentação do produto de tecnologia assistida para obter detalhes sobre como utilizar essas tecnologias com este produto.
- Operem recursos específicos ou equivalentes usando apenas o teclado.
- Ampliem os itens que são exibidos na tela.

Além disso, a documentação do produto foi modificada para incluir recursos que auxiliam na acessibilidade:

- Toda a documentação está disponível nos formatos HTML e PDF que podem ser convertidos para permitir que os usuários apliquem o software de leitor de tela.
- Todas as imagens na documentação são fornecidas com texto alternativo para que os usuários com problemas visuais possam entender o conteúdo das imagens.

Navegando na Interface Utilizando o Teclado

O atalho padrão e as teclas aceleradoras são usados pelo produto e documentados pelo sistema operacional. Consulte a documentação fornecida pelo sistema operacional para obter informações adicionais.

O painel Editor de Regra de Evento é o único que não permite operações apenas pelo teclado e o CSS não pode ser desativado. No entanto, como alternativa, é possível desempenhar todas as operações disponíveis nesse painel, ativando o comando composer a partir da interface da linha de comandos.

Ampliando o Que Está Exibido na Tela

É possível ampliar as informações nas janelas do produto usando os recursos fornecidos pelos sistemas operacionais nos quais o produto está sendo executado. Por exemplo, em um ambiente Microsoft Windows, é possível diminuir a resolução da tela para ampliar os tamanhos das fontes do texto na tela. Consulte a documentação fornecida pelo sistema operacional para obter informações adicionais.

© Copyright IBM Corp. 1999, 2014

Avisos

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos Estados Unidos. É possível que a IBM não ofereça os produtos, serviços ou recursos discutidos nesta publicação em outros países. Consulte um representante IBM local para obter informações sobre produtos e serviços disponíveis atualmente em sua área. Qualquer referência a produtos, programas ou serviços IBM não significa que apenas produtos, programas ou serviços IBM possam ser utilizados. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja nenhum direito de propriedade intelectual da IBM poderá ser utilizado em substituição a este produto, programa ou serviço. Entretanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço não IBM são de responsabilidade do usuário.

A IBM pode ter patentes ou solicitações de patentes pendentes relativas a assuntos tratados nesta publicação. O fornecimento desta publicação não lhe garante direito algum sobre tais patentes. Pedidos de licença devem ser enviados, por escrito, para:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138-146,
Botafogo
Rio de Janeiro, RJ
CEP 22290-240

Para pedidos de licença relacionados a informações de DBCS (Conjunto de Caracteres de Byte Duplo), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos de licença, por escrito, para:

Intellectual Property Licensing
IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106
Japan

O parágrafo a seguir não se aplica ao Reino Unido nem a nenhum país em que tais disposições não estejam de acordo com a legislação local:

A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "NO ESTADO EM QUE SE ENCONTRA", SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS A ELAS NÃO SE LIMITANDO, AS GARANTIAS IMPLÍCITAS (OU CONDIÇÕES) DE NÃO INFRAÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO.

Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, essa disposição pode não se aplicar ao Cliente.

Essas informações podem conter imprecisões técnicas ou erros tipográficos. São feitas alterações periódicas nas informações aqui contidas; tais alterações serão incorporadas em futuras edições desta publicação. A IBM pode, a qualquer momento, aperfeiçoar e/ou alterar os produtos e/ou programas descritos nesta publicação, sem aviso prévio.

© Copyright IBM Corp. 1999, 2014

Referências nestas informações a websites não IBM são fornecidas apenas por conveniência e não representam de forma alguma um endosso a esses websites. Os materiais contidos nesses websites não fazem parte dos materiais desse produto IBM e a utilização desses websites é de inteira responsabilidade do Cliente.

A IBM pode utilizar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Licenciados deste programa que desejam obter informações sobre este assunto com objetivo de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) a utilização mútua das informações trocadas, devem entrar em contato com:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138-146
Botafogo
Rio de Janeiro, RJ
CEP 22290-240

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriadas, incluindo em alguns casos o pagamento de uma taxa.

O programa licenciado descrito nesta publicação e todo o material licenciado disponível são fornecidos pela IBM sob os termos do Contrato com o Cliente IBM, do Contrato Internacional de Licença do Programa IBM ou de qualquer outro contrato equivalente.

Estas informações contêm exemplos de dados e relatórios utilizados nas operações diárias de negócios. Para ilustrá-los da forma mais completa possível, os exemplos incluem nomes de indivíduos, empresas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com os nomes e endereços utilizados por uma empresa real é mera coincidência.

Marcas Registradas

IBM, o logotipo IBM e ibm.com são marcas ou marcas registradas da International Business Machines Corporation nos Estados Unidos e/ou em outros países. Se estes e outros termos de marca registrada da IBM estiverem marcados em sua primeira ocorrência nestas informações com um símbolo de marca registrada (® ou ™), estes símbolos indicarão marcas registradas dos Estados Unidos ou de direito consuetudinário de propriedade da IBM no momento em que estas informações forem publicadas. Estas marcas registradas também podem ser marcas registradas ou de direito consuetudinário em outros países. Uma lista atual de marcas registradas da IBM está disponível na web em "Copyright and trademark information" em http://www.ibm.com/legal/copytrade.shtml.

Adobe, o logotipo Adobe, PostScript e o logotipo PostScript são marcas ou marcas registradas da Adobe Systems Incorporated nos Estados Unidos e/ou em outros países.

Intel, o logotipo Intel, Intel Inside, o logotipo Intel Inside, Intel Centrino, o logotipo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium e Pentium são marcas ou marcas registradas da Intel Corporation ou suas subsidiárias nos Estados Unidos e em outros países.

Microsoft, Windows, Windows NT e o logotipo Windows são marcas registradas da Microsoft Corporation nos Estados Unidos e/ou em outros países.

Java e todas as marcas registradas e logotipos baseados em Java são marcas ou marcas registradas da Oracle e/ou de suas afiliadas.

Linux é uma marca registrada de Linus Torvalds nos Estados Unidos e/ou em outros países.

UNIX é uma marca registrada do The Open Group nos Estados Unidos e em outros países.

Índice Remissivo

A

abend
  estado do fluxo de job 359
  estado do job 352
abenp
  estado do job 352
ação do acionador 18
access
  agentes estendido e de rede 145
  definição de estação de trabalho 145
acessibilidade xiv, 783
Acesso lento ao banco de dados 541
action
  elemento 694
add
  estado do fluxo de job 359
  estado do job 352
afinidade
  definindo 486
  sintaxe 486
afinidade com alias de tarefa 486
afinidade com ID da tarefa 486
afinidade com nome da tarefa 486
agent 10, 13
  definição de estação de trabalho 147
  definindo usuário do Windows 194
  iniciando 37
  método de acesso
    sintaxe 604
  parando 37
agente da rede
  amostra de cenário 618
  definição 617
  dependência de interligação de redes 615
    como criar 619
    gerenciando utilizando o conman 620
  estado EXTERNAL
    ERROR 620
    EXTRN 620
  EXTERNAL 620
  método de acesso
    arquivo de opções 617
  método de acesso netmth 617
  referência 615
  visão geral 615
agente dinâmico
  definição de estação de trabalho 147
  gateway 13
  método de acesso
    arquivo de opções 607
  visão geral 603
Agente do Tivoli Workload Scheduler for z/OS
  método de acesso
    arquivo de opções 607
agente estendido
  definição de estação de trabalho 146
  método de acesso
    arquivo de opções 607
agente estendido *(continuação)*
  método de acesso *(continuação)*
    executando 609
    mensagens de resposta 606
    resolução de problemas 612
  tarefa Check File
    sintaxe 611
  tarefa GS (Get Status)
    sintaxe 611
  visão geral 603
agente padrão
  definição de estação de trabalho 146
agente tolerante a falhas 10
agentes dinâmicos 471, 473
allocation
  elemento 687
alternando agentes estendidos
  palavra-chave $MANAGER 145
  palavra-chave $MASTER 145
ambiente de tarefa no AS/400 647
ambiente de tarefa no i5/OS 647
ambiente de tarefa no IBM i 647
annotation
  elemento 673
API Java 27
aplicativo
  elemento 676
aplicativo de carga de trabalho 272
  definição 3
  linha de comando 335
aplicativos de carga de trabalho
  arquivo de mapeamento 331
  definindo 271
  gerenciando 331
  importando 334
  logs e rastreios 335
  mapeando 334
appservman
  parando 436
argumentos
  elemento 699
arquitetura 31
arquivando
  instâncias de tarefas 541
arquivo CLIConfig.properties
  configuração da linha de comandos 534
arquivo de mapeamento, aplicativos de carga de trabalho 331
arquivo de opções
  agente dinâmico
    método de acesso 607
  Agente do Tivoli Workload Scheduler for z/OS
    método de acesso 607
  agente estendido
    método de acesso 607
arquivo de opções globais
  nome 607
arquivo de opções locais
  nome 607
arquivo Symphony
  JnextPlan 79, 81
  plano de produção 71, 79, 81
arquivos
  at.allow 496
  at.deny 496
  Courier.msg 39
  Intercom.msg 39
  Mailbox.msg 39
  NetReq.msg 38
  PlanBox.msg 39
  Server.msg 39
arquivos da caixa postal
  Courier.msg 39
  definindo o tamanho 509
  Intercom.msg 39
  Mailbox.msg 39
  NetReq.msg 38
  PlanBox.msg 39
  Server.msg 39
Associação de Recurso Lógico
  recuperando 546
associação de tarefas
  definindo 486
ativação de horário de início da previsão 74
ativando
  Comunicação SSL 149
  fuso horário 597
autolink
  definição de estação de trabalho 148
automatizando o processamento
  plano de produção 101
automatizando o processamento do plano
  fluxo de tarefa final 101
autostart monman 449
aviso de recuperação 10

B

banco de dados
  replicando plano 25
behindfirewall
  definição de estação de trabalho 148

C

calendário
  ciclo de execução 10, 209, 212, 229, 245
  dias livres 10
  feriados 10
cancel command 368
candidateCPUs
  elemento 679
candidateHosts
  elemento 678
candidateOperatingSystems
  elemento 681

© Copyright IBM Corp. 1999, 2014

candidateResources
elemento 689
capacidades dinâmicas 471
carryforward
palavras-chave do fluxo de tarefas 71
Personalizando 71
stageman 71
variável 96
carryStates
variável 71, 76
category
elemento 673
chamada de serviço da Web
genérico 473, 487
modelo 474
chamar um serviço da Web 473, 487
arquivos de amostra JSDL 474
ciclo de execução
anualmente 4
baseado em deslocamento 4
baseado em regra 4
calendário 209, 212, 229, 245
data 209, 212, 229, 244
dia 209, 212, 229, 244
diário 4
dias livres 4
em 244
exclusivo 4, 10
icalendar 209, 213, 229, 245
inclusivo 4, 10
semanal 4
simples 4
ciclo de execução anual 4
ciclo de execução com base em deslocamento 4
ciclo de execução com base em regra 4
ciclo de execução de dias livres 4
ciclo de execução diário 4
ciclo de execução exclusivo 4
ciclo de execução inclusivo 4
ciclo de execução semanal 4
ciclo de execução simples 4
ciclo de produção 57
gerenciando 57
identificando instâncias do fluxo de trabalho 60
linha de comandos planman 82
classe
estação de trabalho 14
classe de estação de trabalho 14
código de retorno do usuário no AS/400 646, 648
código de retorno do usuário no i5/OS 646, 648
código de retorno do usuário no IBM i 646, 648
código de retorno no AS/400 646
código de retorno no i5/OS 646
código de retorno no IBM i 646
códigos de retorno
executor de tarefa 476
tarefa com opções avançadas 476
tarefa de serviços da Web 476
tarefa de transferência de arquivos 476
tarefa do banco de dados 476
tarefa Java 476
com recursos de planejamento dinâmico 487
comando
Cpuinfo 612
logman 97
stageman 95
comando add 287
comando adddep job 362
comando adddep sched 364
comando altpass 366
comando altpri 367
comando at 494
variável ATSCRIPT 494
comando authenticate 289
comando batch 494
comando bulk_discovery 368
comando cancel sched 370
comando caxtract 580
comando checkhealthstatus 371
comando confirm 372
comando console 373
comando continue (composer) 290
comando continue (conman) 374
comando cpuinfo 497
comando create 300
comando datamigrate 503
comando datecalc 499
comando deldep job 374
comando deldep sched 376
comando delete 290, 505
comando deployconf 377
comando display 294, 378
comando edit 299
comando evtdef 506
comando evtsize 509
comando exit 299, 380
comando exportserverdata 537
comando extract 300
comando fence 380
comando help 304, 382
Comando importserverdata 539
comando jbxtract 578
comando jobinfo 511
comando jobprop 540
comando jobstdl 513, 518
comando kill 383
comando limit cpu 384
comando limit sched 385
comando link 386
comando list 305
comando listsym 388
comando lock 310
comando maestro 515
comando makecal 516
comando metronome 518, 529
comando modify 314
comando movehistorydata 541
comando new 319
comando param 543
comando parms 520
comando paxtract 580
comando print 305
comando prxtract 579
comando r11xtract 582
comando recall 390
comando redo 321, 391
comando release 522
comando release job 392, 395
comando release sched 393
comando rename 322
comando rep1 561
comando rep11 564
comando rep2 561
comando rep3 561
comando rep4a 561
comando rep4b 561
comando rep7 562
comando rep8 563
comando replace 324
comando reptr 565
comando rerun 396
comando resetFTA 398
comando resource 399, 546
executando a partir do agente
configuração de CLIConfig.properties 554
requirement 554
comando rextract 581
comando rmstdlist 523
comando sendevent 524, 556
comando setsym 400
comando showcpus 401
comando showdomain 408
comando showexec 526
comando showfiles 409
comando showjobs 411
comando showprompts 427
comando showresources 429
comando showschedules 431
comando shutdown 436
comando start 437
comando startappserver 439
comando startbrokerapp 440
comando starteventprocessor 440
comando startmon 441
comando StartUp 527
Comando StartUpLwa 528
comando status 442
comando stop 442
comando stop; progressive 444
comando stopappserver 445
comando stopbrokerapp 447
comando stopeventprocessor 447
comando stopmon 448
comando submit docommand 449
comando submit file 452
comando submit job 456
comando submit sched 459
comando switcheventprocessor 463
comando switchmgr 464
comando tellop 466
comando twstrace 557
comando unlink 466
comando unlock 326
comando validate 329
comando version 330, 469
exibindo o banner do composer 330
Comando wappman 335
logs e rastreios 335
comando xref 567
comando xrxtrct 583
comandos
adddep job 362
adddep sched 364

comandos *(continuação)*
altpass 366
altpri 367
às 494
batch 494
bulk_discovery 368
cancel job 368
cancel sched 370
caxtract 580
checkhealthstatus 371
comando utilitário version 529
confirmar 372
console 373
continue (composer) 290
continue (conman) 374
cpuinfo 497
datamigrate 503
datecalc 499
deldep job 374
deldep sched 376
deployconf 377
display 378
evtdef 506
evtsize 509
excluir 505
executar novamente 396
exit 380
exportserverdata 537
getmon 401
help 382
importserverdata 539
início 437
jbxtract 578
jobinfo 511
jobprop 540
jobstdl 513
kill 383
limit cpu 384
limit sched 385
limite 380
link 386
listsym 388
maestro 515
makecal 516
metronome 518, 529
morestdl 518
movehistorydata 541
param 543
parar 442
parms 520
paxtract 580
prxtract 579
r11xtr 582
recall 390
recurso 399, 546
redo 391
release 522
release job 392
release sched 393
rep1 561
rep11 564
rep2 561
rep3 561
rep4a 561
rep4b1 561
rep7 562
rep8 563
comandos *(continuação)*
reply 395
reptr 565
rextract 581
rmstdlist 523
sendevent 524, 556
setsym 400
showcpus 401
showdomain 408
showexec 526
showfiles 409
showjobs 411
showprompts 427
showresources 429
showschedules 431
shutdown 436
startappserver 439
startbrokerapp 440
starteventprocessor 440
startmon 441
StartUp 527
StartUpLwa 528
status 442
stop ;progressive 444
stopappserver 445, 447
stopeventprocessor 447
stopmon 448
submit docommand 449
submit file 452
submit job 456
submit sched 459
switcheventprocessor 463
switchmgr 464
tellop 466
unlink 466
version 469
xref 567
xrxtrct 583
comandos de relatório 559
Agendamento de Produção Planejada 564
amostra da saída 574
alterando o formato de data 560
amostra da saídas 568
configuração 559
Detalhe da Produção Planejada
amostra da saída 573
Detalhe da Produção Real
amostra da saída 574
Detalhes da Produção Planejada 565
Detalhes da Produção Real 565
Histograma do Job 563
amostra da saída 572
lista de comandos 560
Listagem de Calendários 561
amostra da saída 570
Listagem de Detalhes do Job 561
amostra da saída 568
Listagem de Históricos do Job 562
amostra da saída 572
Listagem de Parâmetros 561
amostra da saída 571
Listagem de Prompts 561
amostra da saída 570
Listagem de Recursos 561
amostra da saída 571
programas de extração 578
comandos de relatório *(continuação)*
caxtract 580
jbxtract 578
paxtract 580
prxtract 579
r11xtr 582
rextract 581
xrxtract 583
Referência Cruzada 567
amostra da saída 576
Resumo da Produção Planejada 565
Resumo da Produção Real 565
comandos de utilitário para agentes dinâmicos
enviando eventos customizados 556
comandos utilitários 493
agentes 533
alterando o formato de data 499
às 494
arquivo at.allow 496
arquivo at.deny 496
variável ATSCRIPT 494
batch 494
configurando variáveis localmente em agentes dinâmicos 540
criando calendários 516
criando e gerenciando variáveis e senhas localmente em agentes dinâmicos 543
definindo eventos customizados 506
definindo o tamanho do arquivo da caixa postal 509
dinâmico 533
enviando eventos customizados 524
excluindo arquivos 505
exibindo a versão do produto 529
exibindo arquivos da lista padrão 523
exibindo jobs em execução 526
exibindo o conteúdo de arquivos da lista padrão 518
gerenciador de domínio dinâmico 533
gerenciando parâmetros localmente 520
getting TWS_home path 515
informações de datamigrate 503
inicializando o netman 527, 528
liberando as unidades de recursos 522
lista de comandos 493
listando arquivos da lista padrão 513
obtendo informações da estação de trabalho 497
obtendo informações do job 511
obtendo relatórios HTML 518, 529
removendo arquivos da lista padrão 523
shutdown 527
ShutDownLwa 527
como criar
aplicativo de carga de trabalho 272
compilador
mensagens 613
composer
linha de comando 26
mensagens 613

computador
recurso 707
computador do computador
recuperando 546
computadores associados a um recurso
recuperando 546
comunicação de rede 40
início do dia 40
processamento de tarefa 40
Comunicação SSL
ativando 149
conexão
definição 625
configuração
relatório de linha de comandos 591
configuração da linha de comandos
arquivo CLIConfig.properties 534
configurações de log no AS/400 643
configurações de log no i5/OS 643
configurações de log no IBM i 643
configurações de tarefa-filha no AS/400 para desempenhos 644
configurações de tarefa-filha no i5/OS para desempenho 644
configurações de tarefa-filha no IBM i para desempenhos 644
configurações de tarefas-filhas para desempenhos no AS/400 644
configurando
propriedades locais 45
conjunto 10, 13, 471
definindo 139
definindo usuário do Windows 194
estação de trabalho 148
conjunto dinâmico 10, 14, 471
definindo 139
definindo usuário do Windows 194
estação de trabalho 148
conjuntos
programando tipos de tarefas com opções avançadas 471, 473
conjuntos dinâmicos
programando tipos de tarefas com opções avançadas 471, 473
conman
linha de comando 26
conman startappserver
JnextPlan 80
connFactory
elemento 705
Console Manager
mensagens 613
convenções, tipo de caractere xiv
convenções de fonte xiv
convenções utilizadas em publicações xiii
Courier.msg 39
cpu
elemento 680
cpuclass
definição de classe de estação de trabalho 153
Cpuinfo
comando 612
cpuname
definição de estação de trabalho 143
CreatePostReports
JnextPlan 80
credencial
elemento 696, 701, 705
criação de tarefa 156
criação de tarefa dinâmica 156
criando previsão
linha de comandos planman 89
criando trial
linha de comandos planman 87
critérios de correspondência
dentro de um intervalo absoluto 63, 239
dentro de um intervalo relativo 62, 239
follows 232
follows absolute to 63
follows previous 62
follows relative to 62
follows sameday 61
mesmo dia 61, 239
precedente mais próximo 62, 239
predecessor 64
predecessor pendente 64
sucessor 64
curingas
composer 281
customização da carga de trabalho
utilizando tabela de variáveis 113
customizando a carga de trabalho
utilizando tabela de variáveis 113

D

d-pool
estação de trabalho 148
dados de plano 25
dano do Symphony
comando resetFTA 398
data
ciclo de execução 209, 212, 229, 244
definição
parâmetros de conexão 53
tabela de variáveis 202
definição da tabela de variáveis
vartable 143
definição de classe de estação de trabalho 153
cpuclass 153
ignorar 154
members 154
definição de domínio 154
ismaster 155
manager 155
parent 155
definição de estação de trabalho 139, 149
access 145
agent 147
agente dinâmico 147
agente estendido 146
agente padrão 146
autolink 148
behinfirewall 148
cpuname 143
domínio 144
fta 146
definição de estação de trabalho *(continuação)*
fullstatus 149
host 145
intermediário 146
manager 146
members 151
os type 143
protocol 150
requisitos 151
secureaddr 144
ServerID 150
tcpaddr 144
timezone 144
tipo 145
definição de regra de evento 260
eventRules.xsd 260
palavras-chave
actionProvider 266
actionType 267
activeTime 263
correlationAttributes 266
descrição 262, 268
eventCondition 263
eventProvider 263
eventRule 262
eventType 264
filteringPredicate 265
Horário de Verão 262
isDraft 262
nome 262
onDetection 267
onTimeOut 267
operator 266
responseType 267
ruleType 262
scope 265, 268
timeInterval 263
timeZone 262
validity 263
definição de tarefa 156
como criar 489, 490
docommand 158
interactive 159
opção de recuperação 160
scriptname 157
streamlogon 159
success condition 159
tarefa 158
tarefas de banco de dados 177
tarefas de comando remoto 183
tarefas de Java 181
tarefas de método de acesso 185
tarefas de serviços da web 166
tarefas de transferência de arquivos 168
tarefas do IBM i 187
tarefas do J2EE 176
Tarefas do MSSQL 179
tarefas do OSLC Automation 188
Tarefas do OSLC Provisioning 190
tarefas do SmartCloud Provisioning 172
tarefas executáveis 182
tarefas JCL 187
tarefas sombra 164
tasktype 159

definição de tarefa *(continuação)*
utilizando variáveis e parâmetros 191
definição do calendário 197
definição do fluxo de tarefa 216
definição do grupo do ciclo de execução 207
definição do parâmetro 198
definição do prompt 10, 204
definição do recurso 206
definição do usuário 192
domínio confiável 195
usando em tipos de tarefa com opções avançadas 195
definindo
dependências
follows 232
needs 243
opens 249
prompts 252
objetos de banco de dados
calendários 197
classes de estações de trabalho 153
domínios 154
estações de trabalho 139
fluxo de tarefas 216
grupo do ciclo de execução 207
prompts 10, 204
recursos físicos 206
regras de eventos 260
serviços da Web 166
tarefas 156
tarefas AS/400 187
tarefas de banco de dados 177
tarefas de Java 181
tarefas de método de acesso 185
tarefas do IBM i 187
tarefas do J2EE 176
Tarefas do MSSQL 179
tarefas do OSLC Automation 188
Tarefas do OSLC Provisioning 190
tarefas executáveis 182
tarefas i5/OS 187
tarefas JCL 187
tarefas sombra 164
transferência de arquivos 168
usuários do Windows 192
variables 198
tarefas de comando remoto 183
tarefas do SmartCloud Provisioning 172
Definindo Agentes em Sistemas AS/400 641
Definindo Agentes em Sistemas i5/OS 641
Definindo Agentes em Sistemas IBM i 641
definindo o aplicativo de carga de trabalho 272
Definindo Objetos
no banco de dados 137
Definindo tarefas do IBMi 641
Definindo tarefas no AS/400 641
Definindo tarefas no i5/OS 641
Definindo tarefas no IBM i 641
dentro de um intervalo absoluto
critérios de correspondência 63
follows 232
follows absolute to 63
dentro de um intervalo relativo
critérios de correspondência 62
follows 232
follows relative to 62
dependência
cruzado 633
entre redes 615, 620
dependência cruzada
como dependência em uma tarefa sombra 629
como incluir no plano 631
definição 629
estação de trabalho de mecanismo remoto 625
etapas para definir 629
fluxo de informações 627, 633
introdução 625
lógica 625
monitorando resolução no plano 631
plano de produção 631
tarefa remota 625
tarefa sombra 625
dependência de interligação de redes
como criar 619
gerenciando utilizando o conman 620
dependência de tarefa
definindo 486
dependências
órfãs 65
dependências cruzadas
definindo 625
gerenciando 625
DESCONHECIDO 159
desempenho do banco de dados
aperfeiçoando 541
destino
elemento 705
destinos da tarefa
definindo 707
dia
ciclo de execução 209, 212, 229, 244
diskSpace
elemento 684
docommand
definição de tarefa 158
domínio 15
definição de estação de trabalho 144
done
estado do job 352
doubleVariable
elemento 675
Dynamic Workload Console
acessibilidade xiv

E

e
elemento 686
editando definições de tarefas 489, 490
educação xiv
ejb
elemento 704
elementos
action 694
allocation 687
annotation 673
aplicativo 676
argumentos 699
candidateCPUs 679
candidateHosts 678
candidateOperatingSystems 681
candidateResources 689
category 673
connFactory 705
cpu 680
credencial 696, 701, 705
destino 705
diskSpace 684
doubleVariable 675
e 686
ejb 704
endpointReference 689
environment 700
estimatedDuration 693
ewlm 692
executável 697
fileSystem 683
groupName 702
grupo 685
hostName 679
invoker 703
j2ee 703
JAASAuthenticationAlias 707
jms 703
jndiHome 704
jobDefinition 672
logicalResource 684
maximumResourceWaitingTime 693
message 705
objective 690
operatingSystem 682
optimization 689
orderedCandidatedWorkstations 678
ou 686
parâmetros 695
password 696, 702, 706
physicalMemory 681
prioridade 693
programando 692
propriedades 685
recoveryActions 694
recursos físicos 676
relacionamento 688
relatedResources 677
requirement 687
script 699
speed 680
stringVariable 674
tpmaction 695
tpmaddress 696
uintVariable 675
userName 696, 701, 706
valor 700
variables 674
variável 700
virtualMemory 681
workflow 697
em
ciclo de execução 244

enCarryForward
variável 71, 75
enCFInterNetworkDeps
variável 76
enCFResourceQuantity
variável 76
endpointReference
elemento 689
enLegacyId
variável 77
enLegacyStartOfDayEvaluation
variável 79, 598
enPreventStart
variável 77
enTimeZone
variável 78, 597
environment
elemento 700
erro
estado do job 352
espelhamento 25
estação de trabalho
agente tolerante a falhas 10
arquivos da caixa postal
NetReq.msg 38
classe 14
como criar 139
conjunto 10
conjunto dinâmico 10
definindo 139
estação de trabalho de mecanismo remoto 10
gerenciador de domínio 10
gerenciador de domínio backup principal 10
gerenciador de domínio mestre 10
processos 31
tipo de conjunto 148
tipo de conjunto dinâmico 148
tipo de d-pool 148
tipo de mecanismo remoto 139, 147
estação de trabalho de mecanismo remoto 10, 14, 20
definindo 625, 629
estações de trabalho dinâmicas 10, 139
estados do fluxo de trabalho
abend 359
add 359
exec 359
pronto 359
stuck 359
succ 359
suspender 359
estados do job
abend 352
abenp 352
add 352
done 352
erro 352
exec 352
extrn 352
fail 352
intro 352
limite 352
pend 352
pronto 352
sched 352
estados do job *(continuação)*
succ 352
succp 353
suspender 352
wait 353
estatísticas da tarefa de criação de log 97
estimatedDuration
elemento 693
eventos customizados
definindo 134, 506
enviando 134, 524
enviando a partir de agentes dinâmicos 556
ewlm
elemento 692
exec
estado do fluxo de job 359
estado do job 352
executando comandos do sistema
a partir do composer 325
a partir do conman 342, 465
executável
elemento 697
extensão de trial
linha de comandos planman 88
EXTERNAL
fluxo de tarefas 620
tarefas 621
extração de dados do banco de dados 473, 474, 487
extrn
estado do job 352

F

fail
estado do job 352
fazendo o upgrade 114
fdignore
em 211, 247
except 214, 231
fdnext
em 211, 247
except 214, 231
fdprev
em 211, 247
except 214, 231
fileSystem
elemento 683
filtros
composer 281
fluxo de tarefa final
automatizando o processamento do plano 101
fluxo de tarefas 2
calculando tempo de execução 100
EXTERNAL 620
FNCJSI
plano de pré-produção 59
follows
critérios de correspondência 232
follows absolute to
critérios de correspondência 63
dentro de um intervalo absoluto 63
follows previous
critérios de correspondência 62
follows previous *(continuação)*
precedente mais próximo 62
follows relative to
critérios de correspondência 62
dentro de um intervalo relativo 62
follows sameday
critérios de correspondência 61
mesmo dia 61
fta
definição de estação de trabalho 146
fullstatus
definição de estação de trabalho 149
fuso horário
ativando 597

G

garantia de serviço de carga de trabalho
calculando horários de início da tarefa 74
plano de previsão 74
gerenciador de domínio 10
gerenciador de domínio backup principal 10
gerenciador de domínio mestre 10
gerenciamento de plano
conceitos básicos 57
logman 97
Personalizando 71, 75
stageman 95
gerenciando
aplicativos de carga de trabalho 271, 331
ciclo de produção 57
critérios de correspondência 61
dependências follows externas 61
objetos no banco de dados 137
tarefa sombra no plano 639
Gerenciando Agentes em Sistemas AS/400 642
Gerenciando Agentes em Sistemas i5/OS 642
Gerenciando Agentes em Sistemas IBM i 642
gerenciando eventos
comutando o servidor de processamento de eventos 463
iniciando o mecanismo de monitoramento 441
iniciando o servidor de processamento de eventos 440
parando o mecanismo de monitoramento 448
parando o servidor de processamento de eventos 447
gerenciando fusos horários 597
nome do fuso horário
com o comprimento da variável 597
gerenciando objetos
no banco de dados 277
no plano 339
gerenciando tarefas e agentes no ambiente dinâmico do IBM i 641
gerenciando tarefas e agentes no AS/400 641

gerenciando tarefas e agentes no i5/OS 641
gerenciando tarefas e agentes no IBM i 641
glossário xiii
groupName
  elemento 702
grupo
  elemento 685
grupo do ciclo de execução 207
  sintaxe da definição de arquivo 713

H

host
  agentes estendidos 145
  definição de estação de trabalho 145
hostName
  elemento 679

I

icalendar
  ciclo de execução 209, 213, 229, 245
identificando instâncias do fluxo de trabalho
  no plano 60
    às 60
    scheddateandtime 60
  plano de pré-produção 60
ignorar
  definição de classe de estação de trabalho 154
informações sobre recurso lógico
  recuperando 546
iniciando
  processos da estação de trabalho 37
  WebSphere Application Server 37
Iniciando e parando agentes em sistemas AS/400 642
Iniciando e parando agentes em sistemas i5/OS 642
Iniciando e parando agentes em sistemas IBM i 642
Iniciando e parando agentes no IBM i 642
iniciando o processamento
  plano de produção 100
início do dia
  estabelecendo comunicação 41
instância do dynamic workload broker
  URI 537, 539
instâncias de tarefas
  arquivando 541
Instruções JSDL 667
Integração com o IBM Tivoli Monitoring 6.1
  bulk_discovery 368
integração EWLM
  ativando em tarefas 689
  capacidade de otimização 689
integridade de dados
  tabela de variáveis 115
interactive
  definição de tarefa 159
Intercom.msg 39
interface 604
interface de linha de comandos
  definição 53
interfaces com o usuário
  API Java 27
  composer 27
  conman 27
  Dynamic Workload Console 27
  Interface de Serviços da Web 28
  optman 27
  planman 28
  plug-ins 27
intermediário
  definição de estação de trabalho 146
intro
  estado do job 352
invoker
  elemento 703
ismaster
  definição de domínio 155

J

j2ee
  elemento 703
JAASAuthenticationAlias
  elemento 707
jms
  elemento 703
jndiHome
  elemento 704
JnextPlan
  conman startappserver 80
  CreatePostReports 80
  MakePlan 80
  SwitchPlan 80
  UpdateStats 80
Job Brokering Definition Console
  editando definições de tarefas 489, 490
job statement
  nos fluxos de job 236
Job Submission Description Language (JSDL) 667
jobDefinition
  elemento 672
jobman
  variáveis de ambiente 46
Jobman
  mensagens 613
JSDL (Job Submission Description Language) 667

L

limit cpu
  processo jobman 33
limite
  estado do job 352
linguagem de planejamento 216
linha de comando
  composer 26
  conman 26
  optman 26
linha de comandos planman
  criando previsão 89
linha de comandos planman *(continuação)*
  criando trial 87
  extensão de trial 88
  monitorar replicação 94
  parâmetros de conexão 82
  plano de desbloqueio 91
  plano de reconfiguração 91
  plano intermediário 84, 85
  recuperando informações do plano 86
  Regras de Implementação 90
  removendo plano 92
  replicando os dados do plano 93
logicalResource
  elemento 684
logmanMinMaxPolicy
  variável 78
logmanSmoothPolicy
  variável 78

M

Mailbox.msg 39
MakePlan
  JnextPlan 80
manager
  definição de estação de trabalho 146
manutenção de tabelas do BD
  comando movehistorydata 541
maximumResourceWaitingTime
  elemento 693
maxLen
  variável 75
mecanismo de bloqueio
  tabela de variáveis 115
mecanismo remoto
  como é conectada 632
  dependência cruzada 633
  estação de trabalho 139, 147
média do tempo de execução 100
melhorando o desempenho do banco de dados 541
members
  definição de classe de estação de trabalho 154
  definição de estação de trabalho 151
mensagens
  compilador 613
  composer 613
  Console Manager 613
  Jobman 613
mesmo dia
  critérios de correspondência 61
  follows 232
  follows sameday 61
message
  elemento 705
método de acesso 604
  agent
    sintaxe 604
  agente dinâmico
    arquivo de opções 607
  Agente do Tivoli Workload Scheduler for z/OS
    arquivo de opções 607
  agente estendido
    arquivo de opções 607

método de acesso *(continuação)*
interface 604
opções de tarefas 604
método de acesso netmth 617
método de acesso para agente dinâmico
visão geral 603
método de acesso para agente estendido
visão geral 603
migrando 114
minLen
variável 75
modelos
para definições de objeto de planejamento 139

N

NetReq.msg 38
nível de log no AS/400 643
nível de log no i5/OS 643
nível de log no IBM i 643
nome
arquivo de opções globais 607
arquivo de opções locais 607
nomes de caminho, notação xv
nomes de diretórios, notação xv
notação
fonte xv
nomes de caminho xv
variáveis de ambiente xv
novo executor 487
novos executores 158, 473
modelo 474
programando 13, 471
tarefa de comando remoto 183
tarefa de serviços da Web 166
tarefa de transferência de arquivos 168
tarefa do SmartCloud Provisioning 172
tarefas AS/400 187
tarefas de banco de dados 177
tarefas de Java 181
tarefas de método de acesso 185
tarefas do IBM i 187
tarefas do J2EE 176
Tarefas do MSSQL 179
tarefas do OSLC Automation 188
Tarefas do OSLC Provisioning 190
tarefas executáveis 182
tarefas i5/OS 187
tarefas JCL 187
novos plug-ins 473, 474, 487
modelo 474
SmartCloud Provisioning 172
tarefa de comando remoto 183
tarefa de serviços da Web 166
tarefa de transferência de arquivos 168
tarefas AS/400 187
tarefas de banco de dados 177
tarefas de Java 181
tarefas de método de acesso 185
tarefas do IBM i 187
tarefas do J2EE 176
Tarefas do MSSQL 179
tarefas do OSLC Automation 188
novos plug-ins *(continuação)*
Tarefas do OSLC Provisioning 190
tarefas executáveis 182
tarefas i5/OS 187
tarefas JCL 187
nuvem
tarefas do SmartCloud Provisioning 172

O

objective
elemento 690
objetos de banco de dados 330
calendários 197
classes de estações de trabalho 153
comando add 287
comando authenticate 289
comando continue 290
comando create 300
comando delete 290
comando display 294
comando edit 299
comando exit 299
comando extract 300
comando help 304
comando list 305
comando lock 310
comando modify 314
comando new 319
comando print 305
comando redo 321
comando rename 322
comando replace 324
comando twstrace 557
comando unlock 326
comando validate 329
Comando wappman 335
domínios 154
estações de trabalho 139
exibindo o banner do composer 330
fluxo de tarefas 216
grupo do ciclo de execução 207
prompts 10, 204
recursos físicos 206
regras de eventos 260
serviços da Web 166
tabela de variáveis 202
tarefas 156
tarefas de banco de dados 177
tarefas de Java 181
tarefas de método de acesso 185
tarefas do IBM i 187
tarefas do J2EE 176
Tarefas do MSSQL 179
tarefas do OSLC Automation 188
Tarefas do OSLC Provisioning 190
tarefas executáveis 182
tarefas JCL 187
tarefas sombra 164
transferência de arquivos 168
usuários 192
variables 198
opção local
variável mm retry link 100
opções de tarefas
método de acesso 604
opções globais
carryStates 71
variável carryforward 96
variável carryStates 76
variável enCarryForward 71, 75
variável enCFInterNetworkDeps 76
variável enCFResourceQuantity 76
variável enLegacyId 77
variável enLegacyStartOfDayEvaluation 79, 598
variável enPreventStart 77
variável enTimeZone 78, 597
variável logmanMinMaxPolicy 78
variável logmanSmoothPolicy 78
variável maxLen 75
variável minLen 75
variável startOfDay 75, 598
operações de transferência de arquivos 473, 487
arquivos de amostra JSDL 474
operações do banco de dados 473, 487
arquivos de amostra JSDL 474
operações Java 473, 487
arquivos de amostra JSDL 474
operatingSystem
elemento 682
optimization
elemento 689
optman
linha de comando 26
orderedCandidatedWorkstations
elemento 678
órfãs
dependências 65
os type
definição de estação de trabalho 143
otimização de recurso
ativando em tarefas 689
integração EWLM 689
otimização de tarefas
integração EWLM 689
otimização EWLM
ativando em tarefas 689
ou
elemento 686

P

palavra-chave $MANAGER 145
palavra-chave $MASTER 145
palavra-chave at 217, 220
palavra-chave carryforward 221
palavra-chave comment 222
palavra-chave confirmed 222
palavra-chave critical 222
palavra-chave deadline 223
palavra-chave description 224
palavra-chave draft 224
palavra-chave end 225
palavra-chave every 225
palavra-chave except 228
palavra-chave follows 232
palavra-chave freedays 235
palavra-chave keyjob 238
palavra-chave keysched 238
palavra-chave limit 238

palavra-chave matching 239
palavra-chave maxdur 240
palavra-chave mindur 241
palavra-chave needs 243
palavra-chave on 244
  ciclo de execução 244
palavra-chave onmaxdur 240
palavra-chave onmindur 241
palavra-chave onuntil 256, 257
palavra-chave opens 249
palavra-chave priority 251
palavra-chave prompt 252
palavra-chave schedtime 253
palavra-chave schedule 254
palavra-chave timezone 256
palavra-chave until 256
palavra-chave validfrom 258
palavra-chave vartable 259
palavras-chave
  a cada 225
  às 217, 220
  carryforward 221
  comentário 222
  confirmed 222
  critical 222
  descrição 224
  dias livres 235
  draft 224
  em 244
  end 225
  except 228
  follows 232
  keyjob 238
  keysched 238
  limite 238
  matching 239
  maxdur 240
  mindur 241
  needs 243
  opens 249
  planejamento 254
  prazo final 223
  prioridade 251
  prompt 252
  schedtime 253
  timezone 256
  until 256
  validfrom 258
  vartable 259
palavras-chave do fluxo de tarefas
  a cada 225
  às 220
  carryforward 71, 221
  comentário 222
  confirmed 222
  critical 222
  descrição 224
  dias livres 235
  draft 224
  em 244
  end 225
  except 228
  follows 232
  job statement 236
  keyjob 238
  keysched 238
  limite 238
palavras-chave do fluxo de tarefas *(continuação)*
  matching 239
  maxdur 240
  mindur 241
  needs 243
  opens 249
  planejamento 254
  prazo final 223
  prioridade 251
  prompt 252
  schedtime 253
  timezone 256
  until 256
  validfrom 258
  vartable 259
palavras-chave dos fluxos de tarefas
  às 217
  onmaxdur 240
  onmindur 241
  onuntil 256, 257
palavras-chave reservadas
  para definições de usuário 139
  para estações de trabalho 138
  para fluxos de tarefas 138
palavras reservadas
  para definições de usuário 139
  para estações de trabalho 138
  para fluxos de tarefas 138
parameter 18
parâmetro global 114
  definição 198
  tabela de variáveis 113
parâmetro local
  banco de dados 520
  definição 198
  exportando 520
  gerenciando 520
  importando 520
parâmetros
  elemento 695
  em definições de job 191
parâmetros de conexão
  definição 53
parâmetros globais 114
parando
  processos da estação de trabalho 37
  WebSphere Application Server 37
parent
  definição de domínio 155
password
  definindo em agente dinâmico 477
  elemento 696, 702, 706
  resolvendo em agente dinâmico 477
  tipos de tarefa com opções avançadas 477
pend
  estado do job 352
physicalMemory
  elemento 681
PlanBox.msg 39
planejamento dinâmico 10, 13, 14, 471
  definição da tarefa 158
  definição de estação de trabalho 139
  definição de tarefa 156
  tipos de tarefas com opções avançadas 471, 474
planejando recurso 18
planman deploy 126
plano
  início rápido 28
plano de desbloqueio
  linha de comandos planman 91
plano de experiência
  como criar 87
  descrição 73
  extensão 88
plano de gerenciamento
  alterando a prioridade 367
  alterando a senha do usuário 366
  alternando o gerenciamento de domínio 464
  atualizando o arquivo de configuração de monitoramento 377
  cancelando fluxos de job 370
  cancelando jobs 368
  confirmando a conclusão do job 372
  definindo o nível da mensagem 373
  designando o console 373
  desvinculando estações de trabalho 466
  encerrando os processos da estação de trabalho 436
  enviando mensagens para o operador 466
  excluindo dependência dos fluxos de job 376
  excluindo dependência nos jobs 374
  executando jobs novamente 396
  executando os comandos novamente 391
  exibindo informações da ajuda 382
  exibindo informações da estação de trabalho 401
  exibindo jobs ou fluxos de job 378
  exibindo o banner do conman 469
  exibindo o status do plano de produção 442
  ignorando o comando 374
  incluindo dependência nos fluxos de tarefa 364
  incluindo dependência nos jobs 362
  iniciando o aplicativo dynamic workload broker 440
  iniciando o servidor de aplicativos 439
  iniciando processos da estação de trabalho 437
  liberando fluxos de job da dependência 393
  liberando jobs da dependência 392
  limitando os jobs em execução em fluxos de job 385
  listando avisos não resolvidos 390
  listando planos processados 388
  modificando a cerca de job 380
  modificando jobs em execução na estação de trabalho 384
  modificando unidades de recursos 399
  mostrando dependências de arquivos 409
  mostrando informações de fluxos de job 431

plano de gerenciamento *(continuação)*
mostrando informações de prompts 427
mostrando informações do domínio 408
mostrando informações do job 411
mostrando informações do recurso 429
obter monitores ativos 401
parando jobs 383
parando o aplicativo dynamic workload broker 447
parando o servidor de aplicativos 445
parando processos da estação de trabalho 442
parando processos da estação de trabalho com proteção do firewall 444
respondendo aos avisos 395
saindo do conman 380
selecionando o plano processado 400
solicitando um bulk_discovery 368
submetendo arquivos como jobs 452
submetendo comandos como jobs 449
submetendo fluxos de job 459
submetendo jobs 456
vinculando estações de trabalho 386
plano de longo prazo
plano de pré-produção 59
plano de pré-produção
descrição 59
FNCJSI 59
plano de longo prazo 59
removendo plano 92
plano de previsão
cálculo do primeiro horário de início 74
como criar 89
descrição 74
plano de produção
arquivo Symphony 57, 79, 81
automatizando o processamento 101
descrição 71
gerando 79, 81
iniciando o processamento 100
JnextPlan 57, 79, 81
monitorar replicação 94
plano de desbloqueio 91
plano de reconfiguração 91
recuperando informações 86
replicando dados 93
plano de reconfiguração
linha de comandos planman 91
plano intermediário
estendendo com 85
gerando 84
plug-ins 27
plug-ins de tarefas de aplicativos 158
programando 13, 471
precedente mais próximo
critérios de correspondência 62
follows 232
follows previous 62
predecessor
critérios de correspondência 64
predecessor *(continuação)*
successor 59, 64
predecessor pendente
critérios de correspondência 64
dependências órfãs 65
successor 64
prioridade
elemento 693
procedimento armazenado do banco de dados
arquivos de amostra JSDL 474
tarefas de banco de dados 473, 487
arquivos de amostra JSDL 474
processamento de tarefa
configurando 52
processo batchman 32
processo de conexão
para tarefa sombra distribuída 632
processo jobman 33
limit cpu 33
processo mailman 32
ServerID 32
processo monman 32
processo netman 32
processo writer 32
processos
batchman 32
jobman 33
mailman 32
monman 32
netman 32
ssmagent 32
writer 32
processos da estação de trabalho 32
árvore de processos no UNIX 34
árvore de processos no Windows 35
batchman 32
comunicação entre os processos 38
gerenciando os estados de alteração de job 42
iniciando 37
início do dia
estabelecendo comunicação 41
jobman 33
mailman 32
ServerID 32
monman 32
netman 32
parando 37
writer 32
programa composer 277
caracteres de controle 280
caracteres especiais 283
códigos de retorno da linha de comandos 283
configuração 277
variables 559
variáveis no UNIX 278
variáveis no Windows 277
curingas 281
delimitadores 283
editor 278
editor XML 278
executando comandos 279
filtros 281
lista de comandos 284
parâmetros de conexão 279
programa composer *(continuação)*
prompt 279
saída do terminal 277
saída off-line 277
programa conman 339
aviso do usuário 342
caracteres de controle 342
caracteres especiais 343
configuração 339
definir variáveis 339
curingas 343
delimitadores 343
executando comandos 341, 343
lista de comandos 360
processando 344
prompt 340
saída do terminal 339
saída off-line 340
selecionando fluxos de tarefas 353
argumentos 354
jobstream_id 354
jobstreamname 354
schedid 354
utilizando at 354
utilizando carriedforward 355
utilizando carryforward 355
utilizando finished 355
utilizando follows 356
utilizando limit 357
utilizando needs 357
utilizando opens 357
utilizando priority 357
utilizando prompt 358
utilizando started 358
utilizando state 358
utilizando until 359
selecionando tarefa 345
argumentos 345
jobstream_ID 346
jobstreamname 345
schedid 346
utilizando at 346
utilizando confirm 347
utilizando critical 347
utilizando critnet 347
utilizando deadline 347
utilizando every 348
utilizando finished 348
utilizando follows 349
utilizando logon 350
utilizando needs 350
utilizando opens 350
utilizando priority 350
utilizando prompt 351
utilizando recovery 351
utilizando scriptname 351
utilizando started 351
utilizando state 352
utilizando until 353
programando
elemento 692
programando tipos de tarefas com opções avançadas 471, 473
promoção de tarefa 104
variáveis de ambiente 486
promoção de tarefa dinâmica 486
promoção de tarefa dinâmica crítica 486

promoção de tarefa em conjuntos dinâmicos 486
promoção de tarefas do broker 486
promovendo uma tarefa 104
prompt
  abend 10
  ad hoc 10
  globais 10
  local 10
  nomeado 10
  recovery 10
prompt abend 10
prompt ad hoc 10
Prompt de comandos do Windows
  nível de privilégio para emitir comandos Tivoli Workload Scheduler 31
prompt global 10
prompt local 10
prompt nomeado 10
pronto
  estado do fluxo de job 359
  estado do job 352
propriedades
  elemento 685
propriedades da estação de trabalho 407
propriedades locais 45
protocol
  definição de estação de trabalho 150
publicações xiii

R

recovery
  definição de tarefa 160
recoveryActions
  elemento 694
recurso
  computador 707
  físico 18
  lógico 18
  programando 18
recurso físico 18
recurso lógico 18
  recurso relacionado 707
recurso relacionado
  recurso lógico 707
  sistema de arquivo 707
  sistema de rede 707
  sistema operacional 707
recursos físicos
  elemento 676
  otimizáveis 708
referência do composer 137
referência do conman 339
regra 18
regra de evento 18
regras de eventos
  cenários de amostra 120, 128
  ocorrências 133
  opção de tempo limite 127
  substituição de variáveis 127
Regras de Implementação
  linha de comandos planman 90
relacionamento
  elemento 688
relacionamento de afinidade
  definindo 486
relatedResources
  elemento 677
relatório de linha de comandos
  configurando 591
relatórios em lote
  amostra de cenário 591
  logs 594
  rastreios 594
rem-eng
  estação de trabalho 147
removendo plano
  linha de comandos planman 92
replicar plano 25
requirement
  elemento 687
requisitos
  definição de estação de trabalho 151
resolução
  variável 116
rrcondsucc
  definição de tarefa 159

S

S.O. Windows
  caracteres especiais, manipulando 98
sched
  estado do job 352
script
  elemento 699
script de configuração .jobmanrc 49
script de configuração djobmanrc 52
script de configuração jobmanrc 48, 51
scriptname
  definição de tarefa 157
scripts de configuração
  .jobmanrc 49
  djobmanrc.cmd 52
  jobmanrc 48
  jobmanrc.cmd 51
secureaddr
  definição de estação de trabalho 144
securitylevel 149
  definição de estação de trabalho 149
segurança
  tabelas de variáveis 116
senha local
  gerenciando em agentes dinâmicos 543
Server.msg 39
ServerID
  definição de estação de trabalho 150
  processo mailman 32
servidor de aplicativos
  parando 445
shutdown
  comando utilitário 527
ShutDownLwa
  comando utilitário 527
sintaxe
  agent
    método de acesso 604
  agente estendido
    tarefa Check File 611
    tarefa GS (Get Status) 611
sistema de arquivo
  recurso relacionado 707
sistema de rede
  recurso relacionado 707
sistema operacional
  recurso relacionado 707
Sistemas operacionais Windows
  nível de privilégio para emitir comandos Tivoli Workload Scheduler 31
speed
  elemento 680
stageman
  carryforward 71
  SwitchPlan 95
startOfDay
  variável 75, 598
status
  atrasado 223
status atrasado 223
status da estação de trabalho 404, 407
status de links da estação de trabalho 404
status de processos da estação de trabalho 404
streamlogon
  definição de tarefa 159
  definição de usuário do Windows 192
stringVariable
  elemento 674
stuck
  estado do fluxo de job 359
succ
  estado do fluxo de job 359
  estado do job 352
successor
  critérios de correspondência 64
  predecessor 59, 64
  predecessor pendente 64
succp
  estado do job 353
Suporte PL/SQL 473
suspender
  estado do fluxo de job 359
  estado do job 352
SwitchPlan
  JnextPlan 80

T

tabela de variáveis 19
  definição 202
  integridade de dados 115
  mecanismo de bloqueio 115
  padrão 114
  utilizando 113
tabela de variáveis padrão
  utilizando 114
tabelas de variáveis
  migração do arquivo de segurança 114
  segurança 116
tarefa 2
  calculando tempo de execução 100
  definição de tarefa 158

tarefa Check File
agente estendido
sintaxe 611
tarefa de ramificação
ações corretivas 758
ações corretivas em sequência 761
avaliando o log da tarefa 742
baseada em ação, definição 757
baseada em condição, definição 733
colocando no fluxo de tarefas 769
comando grep 757
conceitos principais 727
condições complexas 742
diversas ramificações 739
diversas subcondições 754
encerramento anormal do pai 741
especificando no fluxo de tarefas 769
incluindo tarefa de ramificação no banco de dados 766
incluindo tarefa de sinal no banco de dados 766
liberando a ramificação 758
parâmetro IS_CASE_SENSITIVE_i 774
parâmetros fixos 771
parâmetros indexados 771
pausando a ramificação 758
pré-requisitos do Windows 766
pré-requisitos no Windows 766
processo de ação 732
processo de avaliação 731
procura de padrão estendido 747, 749, 751
procurando padrão 743
procurando padrão negativo 745
propagando o status ABEND para o fluxo de tarefas 769
ramificação longa 736
ramificação simples 733
refinando a procura de padrão 757
sinalizando uma ação 763
solicitando confirmação 763
subcondições 742
tarefa de ramificação shell script 766
termos 728
tipos de parâmetros 771
vantagens 732
tarefa-filha no i5/OS 644
tarefa-filha no IBM i 644
tarefa GS (Get Status)
agente estendido
sintaxe 611
tarefa Java genérica 473, 487
modelo 474
tarefa remota
com falha 638
definindo 625
transição de status durante recuperação 638
transportar 639
tarefa sombra 2
com falha 638
definição 164
definindo 625, 629
durante recuperação da tarefa remota 638
tarefa sombra *(continuação)*
gerenciando no plano atual 639
status fail 638
transição de status após conexão 637
transportar 639
tarefas
allocation 707
como criar 707
definindo 707
optimization 707
tarefas
propriedades consumíveis 707
propriedades otimizáveis 707
tarefas antigas
aperfeiçoando 487
tarefas críticas
arquivo de segurança 106
opções globais 104
opções locais 105
tarefas de banco de dados dinâmicas 156
tarefas de Java 473, 487
arquivos de amostra JSDL 474
tarefas de serviço da Web 473, 487
arquivos de amostra JSDL 474
tarefas de serviço da Web dinâmicas 156
tarefas de transferência de arquivos 473, 487
arquivos de amostra JSDL 474
tarefas de transferência de arquivos dinâmicas 156
tarefas dinâmicas 139, 156, 158
tarefa de comando remoto 183
tarefa de serviços da Web 166
tarefa de transferência de arquivos 168
tarefa do SmartCloud Provisioning 172
tarefa MSSQL 179
tarefas AS/400 187
tarefas de banco de dados 177
tarefas de Java 181
tarefas de método de acesso 185
tarefas do IBM i 187
tarefas do J2EE 176
tarefas do OSLC Automation 188
Tarefas do OSLC Provisioning 190
tarefas executáveis 182
tarefas i5/OS 187
tarefas JCL 187
tarefas dinâmicas cruciais 486
tarefas dinâmicas importantes 486
tarefas do banco de dados DB2 177
tarefas do banco de dados MSSQL 177
tarefas do banco de dados Oracle 177
tarefas do IBM i 473, 474
tarefas do J2EE 473, 474
Tarefas do MSSQL 473, 474
tarefas executáveis 473, 474
tarefas existentes
aperfeiçoando 487
tarefas java dinâmicas 156
tarefas padrão
aperfeiçoando 487
tarefas XA 473, 474
tasktype
definição de tarefa 159
tcpaddr
definição de estação de trabalho 144
tempo de execução estimado 100
timezone
definição de estação de trabalho 144
nos fluxos de job 256
tipo
definição de estação de trabalho 145
tipos de recursos
consumível 708
tipos de tarefas 473, 487
modelo 474
tipos de tarefas com opções avançadas 471, 473, 487
arquivos de amostra JSDL 474
definição 2
modelo 474
planejamento dinâmico 2
planejamento estático 2
programando 13, 158
programando dinamicamente 471
programando estaticamente 471
tipos de tarefas específicos 473, 487
arquivos de amostra JSDL 474
tipos de tarefas existentes
definição 2
Tivoli Workload Scheduler
ambiente do tempo de execução 20
arquitetura 31
conceitos básicos 1
controlando o processamento de tarefas 22
definindo atividades 21
emitindo comandos no Windows 31
executando o gerenciamento de eventos 26
gerenciando a produção 25
início rápido 28
interfaces com o usuário 26
objeto 1
processos 31
rede 19
visão geral 1
tpmaction
elemento 695
tpmaddress
elemento 696
transportar
tarefa remota 639
tarefa sombra 639
treinamento
técnico xiv
treinamento técnico xiv
Treinamento Técnico Tivoli xiv
trialsked
plano de experiência 73
plano de previsão 74
TWS_PROMOTED_JOB 610

U

uintVariable
elemento 675
UNIXTASK 159
UpdateStats
JnextPlan 80

Usando Comandos Utilitários em Agentes em Sistemas AS/400 642
Usando Comandos Utilitários em Agentes em Sistemas i5/OS 642
Usando Comandos Utilitários em Agentes em Sistemas IBM i 642
userName
  elemento 696, 701, 706
usuário 19
usuário do Windows
  definição 194
  definindo 194
  executando tarefas em um agent 194
  executando tarefas em um conjunto 194
  executando tarefas em um conjunto dinâmico 194
  planejando em um agent 194
  planejando em um conjunto 194
  planejando em um conjunto dinâmico 194
utilizando
  tabela de variáveis 113
  tabela de variáveis padrão 114

V

validação de dados do banco de dados 473, 474, 487
valor
  elemento 700
variables
  ATSCRIPT 494
  carryforward 96
  carryStates 71, 76
  COMPUTERNAME 46
  elemento 674
  enCarryForward 71, 75
  enCFInterNetworkDeps 76
  enCFResourceQuantity 76
  enLegacyId 77
  enLegacyStartOfDayEvaluation 79, 598
  enPreventStart 77
  enTimeZone 78, 597
  exportadas localmente por .jobmanrc 48, 49, 51, 52
  exportadas no UNIX 46
  HOME 46, 47, 51
  HOMEDRIVE 46
  HOMEPATH 46
  LANG 46, 47
  LD_LIBRARY_PATH 47
  LD_RUN_PATH 47
  LOCAL_RC_OK 48, 51
  logmanMinMaxPolicy 78
  logmanSmoothPolicy 78
  LOGNAME 46, 47
  MAESTRO_OUTPUT_STYLE 46, 47
  MAIL_ON_ABEND 48, 49, 51
  maxLen 75
  minLen 75
  PATH 47
  POSIXHOME 51
  SHELL_TYPE 49
  startOfDay 75, 598
  SystemDrive 46
variables *(continuação)*
  SystemRoot 46
  TEMP 46
  TIVOLI_JOB_DATE 46, 47
  TMPDIR 46
  TMPTEMP 46
  TWS_PROMOTED_JOB 46, 47
  TWS_TISDIR 47
  TZ 46, 47
  UNISON_CPU 46, 47
  UNISON_DATE 46
  UNISON_DATE_FORMAT 47
  UNISON_DIR 46, 47
  UNISON_EXEC_PATH 46, 47
  UNISON_EXIT 48
  UNISON_HOST 46, 47
  UNISON_JCL 48
  UNISON_JOB 46, 47
  UNISON_JOBNUM 46, 47
  UNISON_MASTER 46, 47
  UNISON_RUN 46, 47
  UNISON_SCHED 46, 47
  UNISON_SCHED_DATE 47
  UNISON_SCHED_EPOCH 46, 47
  UNISON_SCHED_IA 46, 47
  UNISON_SCHED_ID 46, 47
  UNISON_SHELL 46, 47
  UNISON_STDLIST 46, 47, 48
  UNISON_SYM 46, 47
  UNISONHOME 46, 47
  USE_EXEC 49
  USERDOMAIN 46
  USERNAME 46
  USERPROFILE 46
  variáveis locais 51, 52, 277
variáveis, ambiente, notação xv
variáveis de ambiente
  promoção de tarefa 486
variáveis de ambiente, notação xv
variáveis de promoção do broker 486
variável 19
  definição 198
  definindo em agente dinâmico 477
  elemento 700
  em definições de job 191
  resolução 116
  resolvendo em agente dinâmico 477
  tipos de tarefa com opções avançadas 477
variável COMPUTERNAME 46
variável HOME 46, 47, 51
variável HOMEDRIVE 46
variável HOMEPATH 46
variável LANG 46, 47
variável LD_LIBRARY_PATH 47
variável LD_RUN_PATH 47
variável local
  gerenciando em agentes dinâmicos 543
variável LOCAL_RC_OK 48, 51
variável LOGNAME 46, 47
variável MAESTRO_OUTPUT_STYLE 46, 47
variável MAIL_ON_ABEND 48, 49, 51
  em uma estação de trabalho com o Windows 51
variável PATH 47
variável POSIXHOME 51
variável SHELL_TYPE 49
variável SystemDrive 46
variável SystemRoot 46
variável TEMP 46
variável TIVOLI_JOB_DATE 46, 47
variável TMPDIR 46
variável TMPTEMP 46
variável TWS_PROMOTED_JOB 46, 47
variável TWS_TISDIR 47
variável TZ 46, 47
variável UNISON_CPU 46, 47
variável UNISON_DATE 46
variável UNISON_DATE_FORMAT 47
variável UNISON_DIR 46, 47
variável UNISON_EXEC_PATH 46, 47
variável UNISON_EXIT 48
variável UNISON_HOST 46, 47
variável UNISON_JCL 48
variável UNISON_JOB 46, 47
variável UNISON_JOBNUM 46, 47
variável UNISON_MASTER 46, 47
variável UNISON_RUN 46, 47
variável UNISON_SCHED 46, 47
variável UNISON_SCHED_DATE 47
variável UNISON_SCHED_EPOCH 46, 47
variável UNISON_SCHED_IA 46, 47
variável UNISON_SCHED_ID 46, 47
variável UNISON_SHELL 46, 47
variável UNISON_STDLIST 46, 47, 48
variável UNISON_SYM 46, 47
variável UNISONHOME 46, 47
variável USE_EXEC 49
variável USERDOMAIN 46
variável USERNAME 46
variável USERPROFILE 46
vartable
  definição da tabela de variáveis 143
verificação de integridade referencial 285
verificando o funcionamento da caixa de correio 371
version
  comando utilitário 529
virtualMemory
  elemento 681
visão geral
  agente dinâmico 603
  agente estendido 603
  método de acesso para agente dinâmico 603
  método de acesso para agente estendido 603

W

wait
  estado do job 353
WAS
  parando 445
WebSphere Application Server
  infra-estrutura 31
  iniciando 37
  parando 37, 445
WINDOWSTASK 159

workflow
elemento 697

X

XATASK 159

Número do Programa: 5698-WSH

Impresso no Brasil

S517-7781-15