

IBM Tivoli Workload Automation



Dynamic Workload Console User's Guide

Version 9 Release 2

IBM Tivoli Workload Automation



Dynamic Workload Console User's Guide

Version 9 Release 2

Note

Before using this information and the product it supports, read the information in “Notices” on page 235.

This edition applies to version 9, release 2, modification level 0 of Tivoli Workload Scheduler (program number 5698-WSH) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 1999, 2014.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures vii

About this publication ix

What is new in this release	ix
Who should read this publication	ix
Publications	ix
Accessibility	ix
Tivoli technical training	ix
Support information	x

Chapter 1. Navigating the Dynamic

Workload Console 1

Naming conventions for scheduling objects	5
Convention for specific platform information	6

Chapter 2. Getting Started 7

Creating and managing engine connections	8
Setting user preferences	8
Event management configuration	9

Chapter 3. Installing and configuring the Dynamic Workload Console 11

Chapter 4. Interact with your environment from a mobile device 13

Chapter 5. Managing users and repositories 15

Managing user settings	15
Changing settings repository	16
Sharing a settings repository	17
Changing the Dynamic Workload Console user of DB repository	18

Chapter 6. Configuring High Availability 19

Changing settings repository	19
--	----

Chapter 7. Customizing your console 21

Customizing your portfolio	21
Customizing your startup page	22
Customizing your tasks	22
Adding tasks to your favorite bookmarks	23
Using the news notification beacon	24
Disabling news notification	24
Customizing the list of notification recipients	25
Sending internal communications	26
Customizing your global settings	27

Chapter 8. Tivoli Workload Scheduler Concepts 31

Scheduling environment	31
Workstation	31

Domain	34
Scheduling objects	37
Job	38
Job stream	38
Workload application	39
Period	39
Calendar	41
Run cycle	41
Run cycle group	43
Operator instructions	47
Parameter	48
Dependencies	48
User	57
Workstation class	57
Variable table	58
Workload Broker job definition	58
Production process	59
Database	59
Plans	60
Preproduction plan	62
Engine connections	63
Event management	64
Reports	66
Workload service assurance	68
Processing and monitoring critical jobs	69
Planning critical jobs	71
Tivoli Workload Scheduler for SAP	73

Chapter 9. Creating and Editing Objects in the Database 75

Designing your Scheduling Environment	75
Creating a workstation	75
Editing workstation definitions	77
Creating a domain	78
Creating a pool of agents	79
Designing your Workload	79
Editing objects from the Working List View	80
Editing objects from the Quick Open View	81
Editing objects from the Details View	82
Editing objects from the Graphical View	82
Editing object properties	83
Creating job stream definitions	83
Creating job definitions	84
Adding a job to a job stream	94
Quick submit of jobs and job streams	96
Adding and removing dependencies	98
Creating cross dependencies	99
Creating and managing run cycle groups and their run cycles	100
Creating Workload Broker objects	107
Creating an event rule	109
Editing event rules	113
Listing object definitions in the database	114
Creating a workload application template	117

Chapter 10. Changing user password in the plan. 121

Chapter 11. Monitoring your Objects in the Plan 123

Monitoring the progress of your plan	123
Monitor tasks	125
Creating a monitoring task query.	125
Monitoring your Scheduling Environment.	127
Creating a task to Monitor Workstations	127
Creating a task to Monitor Domains.	129
Monitoring your Workload.	130
Event management tasks	131
Plan view tasks.	132
Creating a task to Monitor Jobs	132
Creating a task to Monitor Critical Jobs.	134
Creating a task to Monitor Tasks on Multiple Engines	137
Creating a task to Monitor Job Streams.	138
Creating a task to Monitor Job Streams on Multiple Engines	140
Creating a task to Monitor Files	141
Creating a task to Monitor Prompts	143
Creating a task to Monitor Resources	144
Creating a task to Monitor Event Rules.	145
Creating a task to Monitor Triggered Actions	147
Creating a task to Monitor Operator Messages	148
Controlling Jobs and Job Streams Processing	149
Using dependencies to control job and job stream processing	149
Using time restrictions to control job and job stream processing	150
Using job priority and workstation fence to control distributed job processing.	151
Using limits to control job and job stream processing	151
Using job confirmation to control job processing	152
Using job recovery actions to control job processing	152

Chapter 12. Working with Plans 155

Selecting the working plan	155
Generating Trial and Forecast Plans	157
Creating a task to Show Plan View	157
View preproduction plan	159

Chapter 13. Submitting Workload on Request in Production 161

Submitting ad hoc jobs	161
Submitting predefined jobs.	161
Submitting predefined job streams	162
Setting properties for ad hoc jobs and predefined jobs and job streams	162

Chapter 14. Reporting. 165

Creating a task to generate a Job Run Statistics report	165
Creating a task to generate a Job Run History report	166

Creating a task to generate a Workstation Workload Summary report	167
Creating a task to generate a Workstation Workload Runtimes report	168
Creating a task to Create Plan Reports	168
Creating a task to Create Custom SQL Reports	170

Chapter 15. Scenarios. 171

Customizing your job stream	171
Using workload service assurance to monitor z/OS critical jobs	173
Monitoring jobs running on multiple engines.	175

Chapter 16. Troubleshooting the Dynamic Workload Console 181

Chapter 17. Reference 183

Accessing online publications	183
Dynamic Workload Console global settings	183
Override graphical view limits	184
Plan View in new window	185
Disable and customize NewsFeed function	185
Disable and customize the creation of predefined tasks	187
Add customized URL to job and job streams	187
User registry	189
z/OS http connections	190
Limit the number of objects retrieved by queries	190
Limit task and engine sharing	191
Entries in Workload Designer search windows	192
Show all dependencies	192
Auditing mobile app activity	193
TdwcGlobalSettings.xml sample	194
Users and groups	196
Type of communication based on SSL communication options	198
Status description and mapping for distributed jobs	199
Status description and mapping for z/OS jobs	201
Status description and mapping for distributed job streams	203
Status description and mapping for z/OS job streams	205
Workstation types	206
Graphical Views in the plan	210
Workload Designer	214
Working List	214
Quick Open	215
Details view.	216
Graphical View.	218
Run Cycle Preview	221
Message History	222
Using recovery options in job definitions	222
Reports	224
Regular Expressions	225
SQL report examples	229
Event rule	231
Action properties	231
Event properties	232
Activation statuses of the event rules	233
Schemas for generic events	233

Notices	235
Trademarks	236

Index	239
------------------------	------------

Figures

1. Single-domain network	35	6. Example of recovery job with condition dependencies	55
2. Multiple-domain network.	36	7. Cross dependencies	57
3. Example of a condition dependency definition	53	8. Critical path	70
4. Example of a condition dependency at run time	53		
5. Auto-recovery job stream with step level dependency	54		

About this publication

IBM® Tivoli® Workload Scheduler simplifies systems management across distributed environments by integrating systems management functions. Tivoli Workload Scheduler plans, automates, and controls the processing of your enterprise's entire production workload. The *Tivoli Workload Scheduler Dynamic Workload Console User's Guide* provides detailed information about how to configure and use the Dynamic Workload Console to manage your Tivoli Workload Scheduler environment.

What is new in this release

For information about the new or changed functions in this release, see the *IBM Tivoli Workload Automation Overview*, section *Summary of enhancements*.

For information about the APARs that this release addresses, the Dynamic Workload Console Release Notes at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27041033>.

Who should read this publication

This publication is intended for the following audience:

- Tivoli Workload Scheduler operators
- Tivoli Workload Scheduler administrators

Publications

The Tivoli Workload Automation product is supported by a set of publications.

For a list of publications in the Tivoli Workload Automation product library, see *Publications* under *Reference* in the product documentation.

For a list of terms used in the Tivoli Workload Automation product, see *Glossary* under *Reference* in the product documentation.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully.

With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

For full information with respect to the Dynamic Workload Console, see the Accessibility Appendix in the *IBM Tivoli Workload Scheduler User's Guide and Reference*.

Tivoli technical training

Tivoli provides technical training.

For Tivoli technical training information, refer to the following IBM Tivoli Education website:

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

Support information

IBM provides several ways for you to obtain support when you encounter a problem.

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

- Searching knowledge bases: You can search across a large collection of known problems and workarounds, Technotes, and other information.
- Obtaining fixes: You can locate the latest fixes that are already available for your product.
- Contacting IBM Software Support: If you still cannot solve your problem, and you need to work with someone from IBM, you can use a variety of ways to contact IBM Software Support.

For more information about these three ways of resolving problems, see the appendix on support information in *Tivoli Workload Scheduler: Troubleshooting Guide*.

Chapter 1. Navigating the Dynamic Workload Console

For an interactive overview of the product and its features, you can view several demo scenarios, available (in English only) in the Tivoli Workload Scheduler Wiki Media Gallery.

Note: To optimize the viewing of the demos, ensure that you have the latest version of Adobe Flash Player installed.

To have a quick and rapid overview of the portal and of its use, after logging in to Dashboard Application Services Hub (DASH), click one of the hyperlinks displayed in the welcome page to launch videos and a tutorial that helps you find the information you need.

Several products can be integrated in this portal and their related entries are listed together with those belonging to the Dynamic Workload Console in the toolbar displayed on the left side. Use these icons to perform your tasks.

In the following sections only pages and tasks relating to the Dynamic Workload Console are described.

Use the toolbar on the left side to work with the Dynamic Workload Console to perform the following tasks:



- Click the **Search** icon and enter whole or part of the name of the page you want to open. Search results are dynamically returned and filtered as you type.



- Click the **Favorite** icon to create your customized portfolio of pages. To add a task to Favorites, drag it to the Favorites icon. Entries previously displayed in the My Tasks list have been automatically added to Favorites.

The icons displayed in the central part of the toolbar are product-specific and are described in detail in the following section.

The icons located in the bottom part of the toolbar relate to console tasks, for view management, console administration, and help:



- Click the **Users** icon to personalize the console, manage your security settings, or log out. Add, remove, and reorder your favorite tasks. Personalize which pages automatically open when you log in with the My Startup Pages task. Manage your credentials used to access other systems or data providers with the Credential Store task.



- Click the **Views** icon to change the current view. Views control which product tasks or pages are displayed and which pages open automatically.



- Administrators use the **Console Settings** icon to manage settings. The General section contains tasks to manage pages, create a connection to a data provider, create widgets, and export the console setting for migration and backup. Views and console preference profiles personalize the console for various roles making it more usable. The Role section contains tasks to create and assign roles to users and groups.



- Click the **Help** icon to access help contents for the console and the installed product:
 - InfoCenter. View, browse, search, and print online product information. Collaborate with other information center users by adding comments.*
 - Dashboard Community. Consult best practices, videos, the image gallery, the message board, and the blog on the IBM Dashboard Application Services Hub community.*
 - Service Management Connect. Connect, learn, and share with the Service Management community on Service Management Connect.*
 - Welcome. Access multimedia tutorials and videos, and the Dashboard Application Services Hub community.*
 - About. Find the release level of the console and installed product.*

Administration - Create and modify workload definitions in the database



- Click the **Administration** icon to perform tasks that are run typically by the product administrator.

Workload Design

You can manage your workload to design and edit objects in the database, to handle events and SAP jobs.

To create and edit the definitions of workload objects in the database, click **Manage Workload Definitions** and open the Workload Designer. From the Workload Designer, you can create and edit the following objects:

Table 1. Workload Designer objects

Distributed environment	z/OS environment
Jobs	Jobs
Job Streams	Job Streams
Prompt	Periods
Resources	Calendars
Users	Variable Tables
Calendars	Operator Instructions
Workstation Classes	Event Tracking Criteria
Variable Tables	Run Cycle Groups
Workload Application templates	
Run Cycle Groups	

Open the event related pages to create a new event rule or to manage the

list of all the event rules associated to an engine, and modify their definitions.

Use this section also to manage jobs that run on SAP systems.

Submit jobs and job streams not included in the plan:

Workload Submission

Use this section to find and submit jobs and job streams that are already defined in the database, but are not included in the current plan.

Design and modify workstation and domain definitions in the database:

Use this section to design and control the topology of your scheduling environment, that is, your workstations and domains.

Workload Environment Design

Open the pages available under this section to create new workstation definitions, associate them to domains, list all existing workstations associated to an engine, and modify their definitions.

You can also temporarily (for current session only) modify the user password associated to the current plan.

Design and modify Broker jobs and resources:

Broker Design

Use this section to create new jobs on Broker systems and to modify existing jobs. From this section, you can also create and manage Broker logical resources and resource groups.

Planning - View and manage plans



Click the **Planning** icon to perform planning-related tasks.

Workload Forecast

Use this section to work with plans, creating and viewing trial and forecast plans and listing archived plans. From this section you can also view your preproduction plan.

Reporting - Define and run reports



Click the **Workload Reporting** icon to define and run reports to gather historical data or details about your plans.

Workload Reporting

Use this section to create and run reports about your workload. You can also generate and run customized SQL reports.

System Configuration - Create and manage engines and configuration settings



Click the **System Configuration** icon to create new engine connections and manage the existing ones. From this section you can also configure your preferences (time zone, lines per page, layout, dashboard) as well as your settings repository.

Workload Scheduling Settings

Use this section to create and edit your engine connections and your user preferences. Use it also to import and export settings like user preferences, configured tasks, and engine connections to a local file or external repository database. If you want to use a repository database (to share it with other Dynamic Workload Console instances in high availability configuration), you must specify it in this section.

Broker Settings

Use this section to manage Broker server connections and user preferences.

System Status and Health - Create and run tasks to monitor workload and environment



Click the **System Status and Health** icon to define and run customized tasks (reusable queries) and predefined tasks to monitor your workload processing. From this section you can open the Dashboard to view the workload processing in a graphical format on several engines at the same time. From the dashboard, you can browse to view more detailed job information.

Create and run monitor tasks to obtain a list of objects on which you can perform monitoring and control actions. Creating and running a monitor task means creating a filter and running a search on it. From this list, you can click an item to view its properties or to run actions against it. In this way you can easily change some settings and the processing of the plan.

Workload Monitoring

Use this section to create tasks to monitor jobs (on single or multiple engines), critical jobs, job streams, files, resources, prompts and to open a graphical plan view. The result of your queries on jobs and job streams can also be represented in a graphical view.

Environment Monitoring

Use this section to create tasks to monitor workstations and domains.

Event Monitoring

Use this section to create tasks to monitor event rules, operator messages, and the actions triggered by the event rules.

Broker Monitoring

Use this section to monitor jobs, computers, logical resources, and resource groups on Broker systems.

Troubleshooting and Support - Access support information



Click the **Troubleshooting and Support** icon to access support resource. The **Workload Support Landing Page** is your first stop when looking for help on the Tivoli Workload Scheduler product.

Related tasks:

“Accessing online publications” on page 183

Accessing the Tivoli products online publications in the Tivoli software information center website.

Related information:

Chapter 15, “Scenarios,” on page 171

Naming conventions for scheduling objects

The Dynamic Workload Console allows you to manage and control Tivoli Workload Scheduler production for z/OS and distributed environments.

There are some differences in the processing and behavior between the Tivoli Workload Scheduler products for z/OS and distributed environments. When there are differences, the descriptions and related actions of scheduling objects are explained for both environments.

Table 2 lists the objects and object names typical of the Tivoli Workload Scheduler environment where they are defined.

Table 2. Naming convention for scheduling objects

Object description	Object name in a distributed environment	Object name in a z/OS environment
An ordered list of activities in plan for the current production period.	Production Plan	Current Plan
A unit of work that is part of an application or a job stream and that is processed at a workstation.	Job	Operation. An operation can contain a list of steps to run.
A list of jobs that run as a unit to accomplish a task (such as calculating payroll), together with times, priorities, and other dependencies that determine the order in which the jobs run.	Job stream	Application
A run of a job stream or an application scheduled in the plan.	Instance	Occurrence

Table 2. Naming convention for scheduling objects (continued)

Object description	Object name in a distributed environment	Object name in a z/OS environment
A type of application description related to run cycle, calendar information, or job descriptions common to all applications defined as members of the group.	N/A	Application Group
A physical or logical asset where job processing occurs.	Workstation. It is qualified according to its position in the topology of the scheduling network and on its ability to interact with the information contained in the current plan.	Workstation. It is qualified according to the type of job processing it does in computer workstation, general workstation, print workstation.
Tivoli Workload Scheduler database	A customized set of tables in a relational database containing definitions for all scheduling objects, network topology, variables, and job processing statistics.	A collection of six sets of data, acting as a flat database, that contain information about calendars, periods, workstation descriptions, JCL variable tables, application descriptions, and operator instructions.

Related concepts:

“Workstation” on page 31

“Job” on page 38

“Job stream” on page 38

“Production process” on page 59

Convention for specific platform information

Icons to identify the information related only to specific platforms.

This publication uses the following icons to identify the information related only to specific platforms:

Distributed

The information applies only to Tivoli Workload Scheduler running in a distributed environment.

z/OS

The information applies only to Tivoli Workload Scheduler running in a z/OS environment.

All information that is not marked by an icon applies to all the supported environments.

Chapter 2. Getting Started

Information about the Dynamic Workload Console installation and configuration

For more information about this installation, see [Planning and Installation](#) or [Planning and Installation](#)

To configure the Dynamic Workload Console, see [Configuring the Dynamic Workload Console](#) to find information about:

- [Launching in context with the Dynamic Workload Console](#)
- [Configuring access to the Dynamic Workload Console](#)
- [Configuring Dynamic Workload Console to use Single Sign-On](#)
- [Configuring the use of Lightweight Third-Party Authentication](#)
- [Configuring Dynamic Workload Console to use SSL](#)
- [Customizing your global settings](#)
- [Configuring Dynamic Workload Console to view reports](#)

You can access the Dynamic Workload Console from any computer in your environment using a web browser through either the secure HTTPS or HTTP protocol.

When you log in to the Dashboard Application Services Hub, you see a portfolio on the left containing an entry for each Tivoli product installed and integrated with it, for example, the Dynamic Workload Console. Expand the Tivoli Workload Scheduler entry in the portfolio to start to work with the console.

To access the Dynamic Workload Console online embedded documentation, click **Help** at the top right of the panel and select IBM Tivoli Workload Scheduler in the Contents section on the left of the panel that opens.

The first and main actions that you perform when you connect to the Dynamic Workload Console are:

Creating a connection to a Tivoli Workload Scheduler engine

You specify the details (such as IP address, user name, and password) to access a Tivoli Workload Scheduler engine, and, optionally, a database to operate with objects defined in plans or stored in the database.

From the Dynamic Workload Console, you can access the current plan, a trial plan, a forecast plan, or an archived plan for the distributed environment or the current plan for the z/OS environment.

You might want to access the database to perform actions against objects stored in it or to generate reports showing historical or statistical data.

In addition, working both on the database and on plans, you can create and run *event rules* to define and trigger actions that you want to run in response to events occurring on Tivoli Workload Scheduler nodes.

Defining a scheduling environment

You define your Tivoli Workload Scheduler network. You create workstation definitions in the database to represent the physical machines or computer systems on which your workload is scheduled to run. A Tivoli Workload Scheduler network is made up of the workstations where job

and job stream processing occurs. When you design your network, you assign roles to these workstations to suit your specific business requirements. You can design your network with multiple domains, to divide control of a large network into smaller manageable groups. A typical Tivoli Workload Scheduler network consists of a workstation acting as the master domain manager and at least one domain.

Defining scheduling objects in the database

You define your workload, which consists of jobs that are concatenated in job streams. Then, you specify the calendars and run cycles according to which job streams must run. You can also define dependencies to condition the workload processing. All these definitions can be done within the Workload Designer.

Creating tasks to manage Tivoli Workload Scheduler objects in the plan

You specify some filtering criteria to query a list of scheduling objects whose attributes satisfy the criteria you specified. Starting from this list, you can navigate and modify the content of the plan, switching between objects, opening more lists, and accessing other plans or other Tivoli Workload Scheduler environments.

Related concepts:

“Designing your Scheduling Environment” on page 75

“Monitoring your Scheduling Environment” on page 127

“Monitoring your Workload” on page 130

Related tasks:

“Creating and managing engine connections”

“Designing your Workload” on page 79

Creating and managing engine connections

To create, modify, or delete an engine connection, perform the following steps.

Note: You can modify or delete only engine connections that you have created.



1. From the navigation toolbar, click **System Configuration > Manage Engines**.
2. From the displayed panel you can create, edit, delete, or share an engine connection, and test the connection to the remote server where Tivoli Workload Scheduler is installed. You can order the list of engine connections displayed in this panel by using sorting criteria that you select with the buttons at the top left corner of the table.

Related concepts:

“Scheduling objects” on page 37

“Engine connections” on page 63

Setting user preferences

To set the preferences to use in the output of tasks and reports, perform the following steps.

Note: The preferences that you set in this panel are not used in the output of plan reports. Those reports follow the preferences set on the workstation where the query is run.



1. From the navigation toolbar, click **System Configuration > Set User Preferences**. The Set User Preferences panel opens, containing the current settings.
2. To modify the current settings click **Edit**. In this panel you can specify the:
 - Number of rows that are displayed in the table of results of all your tasks as the default setting
 - Options to display dates, times, and time zones
 - Layout and the refresh rate for the dashboard
3. Click **Save changes** to save the changes you made or **Discard changes** to exit the page without saving.

In the Manage User Preferences panel, you can also enable again the news notification. For details, see “Disabling news notification” on page 24.

Event management configuration

You can use the event management feature both from the Tivoli Workload Scheduler command line interface and from the Dynamic Workload Console.

You need the following authorizations to perform event management operations from the Dynamic Workload Console:

On Dashboard Application Services Hub

The user ID you use to log in to Dynamic Workload Console must be defined as user on Dashboard Application Services Hub and must be defined within one of the following groups:

Table 3. Event Management Authorizations

Groups	Event management operations you can perform
TWSWEBUIOperator	List and manage Event Rule Instances, Log Messages, and Triggered Actions.
TWSWEBUIDeveloper	Create, list, and manage Event Rules.

Note: Dynamic Workload Console users belonging to the **TWSWEBUIAdministrator** group can perform all operations available in the web-based user interface.

On Tivoli Workload Scheduler

The Tivoli Workload Scheduler user credentials defined in the engine connection must belong to a Tivoli Workload Scheduler user authorized to perform event management operations in the Tivoli Workload Scheduler security file.

You need the *create* permission set for the *rule* object. You also need the *use* permission on the objects (job, job stream, and so on) that you want to use as events.

For more information about how to define and manage user authorizations in the security file, see Administration Guide.

Related concepts:

“Event management” on page 64

Chapter 3. Installing and configuring the Dynamic Workload Console

For more information about this installation, see Planning and Installation or Planning and Installation

To configure the Dynamic Workload Console, see Configuring the Dynamic Workload Console to find information about:

- Launching in context with the Dynamic Workload Console
- Configuring access to the Dynamic Workload Console
- Configuring Dynamic Workload Console to use Single Sign-On
- Configuring Dynamic Workload Console to use SSL
- Customizing your global settings
- Configuring High Availability for Dynamic Workload Console
- Configuring Dynamic Workload Console to view reports

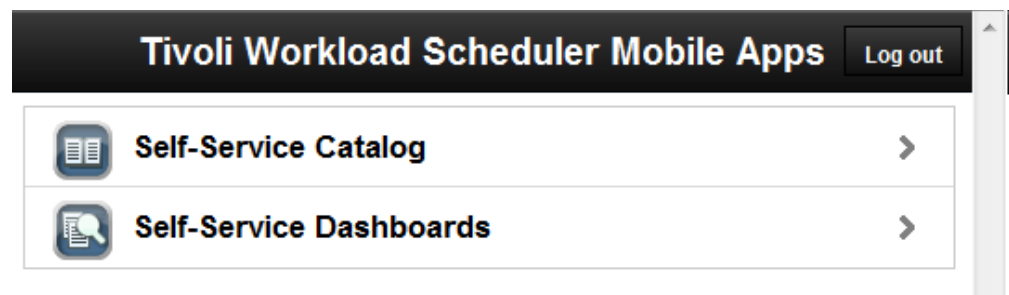
For more information about z/OS connector instances configuration by using WebSphere Application Server tools, see *Tivoli Workload Scheduler for z/OS Planning and Installation* > *Tivoli Workload Scheduler for z/OS connector* > *Installing, Upgrading and Uninstalling Tivoli Workload Scheduler for z/OS connector on distributed systems*.

Chapter 4. Interact with your environment from a mobile device

Use your mobile device to easily and quickly interact with your Tivoli Workload Scheduler environment. You can monitor and run your workload from your mobile device by simply logging in to a URL and starting an app.

The IT market is moving towards mobile devices, which help you perform a large number of tasks, such as manage your sales workforce, read your email, check your accounting system, or attend a web conference. Applications designed for mobile devices must be intuitive and user-friendly while remaining robust and reliable, and providing instant access to business and client data wherever they are.

You can interact with Tivoli Workload Scheduler by using the following applications:



To open this home page on your mobile device, access the following URL:

`https://host_name:port_number/ibm/TWSWebUI/mobile.jsp`

where *host_name* and *port_number* are the host name and port number of the Dynamic Workload Console you are connecting to.

Self-Service Catalog

Define services that correspond to Tivoli Workload Scheduler job streams and submit them from your mobile, even if you do not have any experience with Tivoli Workload Scheduler. Services are organized into catalogs. Launch the Self-Service Catalog from your mobile device by connecting to the following URL:

`https://host_name:port_number/ibm/TWSWebUI/sscatalog.jsp`

where *host_name* and *port_number* are the host name and port number of the Dynamic Workload Console you are connecting to.

To launch and use this application, you must have one of the following roles:

TWSWEBUIAnalyst

This is the minimum role required to access Self-Service Catalog. Users with this role can view catalogs and services to which they are authorized and submit service requests. They cannot modify services or catalogs.

TWSWEBUIAdministrator

Users with this role can create, edit, and delete catalogs and

services. They can also associate roles to services and catalogs to authorize other users to work with them.

Self-Service Dashboards

By defining filter criteria to be applied to your jobs and workstations, you can view dashboards and drill down to more detailed information about the jobs and workstations that match the criteria. You can also perform recovery actions on the jobs and workstations. Launch the Self-Service Dashboards app from your mobile device by connecting to the following URL:

`https://host_name:port_number/ibm/TWSWebUI/ssmanagement.jsp`

where *host_name* and *port_number* are the host name and port number of the Dynamic Workload Console you are connecting to.

To launch and use this application, you must have one of the following roles:

TWSWEBUIAnalyst

This is the minimum role required to access Self-Service Dashboards. Users with this role can view dashboards for which they are authorized but they cannot modify dashboards.

TWSWEBUIAdministrator

Users with this role can create, edit, and delete dashboards. They can also associate roles to dashboards to authorize other users to work with them.

Related concepts:

“Monitoring the progress of your plan” on page 123

Request a graphical view of the progress of the current plan.

Chapter 5. Managing users and repositories

How to configure, change, and share your settings repository and change the DB2 user. How to manage the user settings.

The user settings such as user preferences, saved tasks, and engine connections, which by default are stored in a settings repository that is a local XML file, must be exported and stored in a settings repository on a DB2 database. Using a database as your repository, all your existing user settings relating to current Dynamic Workload Console are saved in the database, and all the operations involving user settings are run using the settings in this repository.

Managing user settings

How to export the user settings and import them into a new Dynamic Workload Console

To perform this task you need to have the TWSWEBUIAdministrator role.

User settings such as user preferences, saved tasks, and engine connections are stored in the settings repository, which by default is a local file. However, you can decide to have your settings repository on a database for all Dynamic Workload Console operations that involve user settings.

You can export the content of your settings repository as an XML file, optionally modify it, and then import it into the same or another instance of Dynamic Workload Console.

This is particularly useful for migration purposes or if you want to modify the same settings in multiple Dynamic Workload Console instances.

To export the settings and import them into a new Dynamic Workload Console, perform the following procedure.

Note: Import and export operations are performed from and to the currently-selected repository.



1. From the navigation toolbar, click **System Configuration > Manage Settings**.
2. In the Manage Settings panel, click **Export Settings** and save the XML file in a directory of your choice.
3. Optionally, edit the file using an XML editor and save it.
4. Log in to the Dynamic Workload Console where you want to import the settings and open the Manage Settings panel.
5. Click **Import Settings** and browse to the XML file containing the settings you want to import. During the import operation, you choose to update or to overwrite the existing settings with the new settings.
6. If you are using a local file as repository, restart the Dynamic Workload Console to make the change effective or click **Undo** before restarting it to restore previous settings. This is not required if you are using a database as

your repository, but in this case you must ensure that during the import operation there are no other users connected to the Dynamic Workload Console while the repository content is being updated.

Changing settings repository

Changing the settings repository.

- To perform this task you need to have the TWSWEBUIAdministrator role.
- You need to have access to an installed DB2 where a database has already been created. If you need information about how to create a DB2 database, see IBM DB2 Database for Linux, UNIX, and Windows Information Center.
- You must have Administrator rights on that database.

User settings such as user preferences, saved tasks, and engine connections are stored in the settings repository, which by default is a local file. However, you can decide to have your settings repository on a database for all Dynamic Workload Console operations that involve user settings.

This can be useful, for example, for scalability purposes or to have multiple Dynamic Workload Console instances sharing the same user settings.

To use a database for your settings repository, you must configure the database settings, as described in the following procedure:

1. Run the wastool:
 - a. From the Tivoli Workload Scheduler installation path, open the `install_dir\wastools` directory and run `installTDWCDataSource wastool` to create the data source.
 - b. Specify the connection details of the DB2 database in `TDWCDataSource.properties` file, located in `install_dir\wastools` directory.
2. Restart the Dynamic Workload Console.
3. Export your settings:



- a. From the navigation toolbar, click **System Configuration > Manage Settings**.
 - b. Optionally, in the Manage Settings panel, click **Export Settings** and save the XML file in a directory of your choice. In this way you save your user settings in a local file to load them on the database, when it becomes your settings repository.
4. Switch repository to DB2.
 - a. In the same panel, click **Configure settings repository > Use database as settings repository** to specify that settings must be saved in the database instead of a local file.
 - b. In the **Database connection properties** section, specify the credentials required to connect to the database.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.

- c. Optionally, you can test the connection.
 - d. Save the new configuration.
5. Import your settings or initialize the database:

- a. Optionally, click **Import settings** to import your user settings from the XML file to the database repository. During the import operation, keep the default choice, which overwrites the existing settings with the new settings. Performing this step, the database is automatically initialized.
- b. If you have not performed previous step, click **Configure settings repository > Initialize database**.

As a result, all your existing user settings relating to current Dynamic Workload Console are saved in the database, and all the operations involving user settings are run using the settings in this repository.

Sharing a settings repository


How to share a settings repository on multiple Dynamic Workload Console instances.

To perform this task you need to have the TWSWEBUIAdministrator role.

User settings such as user preferences, saved tasks, and engine connections are stored in the settings repository, which by default is a local file. However, you can decide to have your settings repository on a database for all Dynamic Workload Console operations that involve user settings.

This can be useful, for example, for scalability purposes or to have multiple Dynamic Workload Console instances sharing the same user settings.

To use a database for your settings repository, you must configure the database settings, as described in the following procedure:

1. Ensure that all the Dynamic Workload Console instances that will share the same settings repository, also use the same user registry.
2. Ensure that one Dynamic Workload Console settings repository has been switched to database, as described in Switch repository to DB2.
3. Specify this repository as the default one for all the other Dynamic Workload Console instances that must share it:
 - a. From the Tivoli Workload Scheduler installation path, open the `install_dir\wastools` directory and run `installTDWCDataSource wastool` to create the data source, specifying the same database settings as in the first Dynamic Workload Console instance.
 - b. Restart the Dynamic Workload Console.
- c. From the navigation toolbar, click  **System Configuration > Manage Settings**.
- d. In the same panel, click **Configure settings repository > Use database as settings repository** to specify that settings must be saved in the database instead of a local file.
- e. In the **Database connection properties** section, specify the credentials required to connect to the database.
- f. Optionally, you can test the connection.
- g. Save the new configuration.

As a result, all user settings are saved in the database, shared by all the Dynamic Workload Console instances, and all the operations involving user settings are run using the settings in this repository.

Changing the Dynamic Workload Console user of DB repository

How to change the Dynamic Workload Console user that updates the settings repository on DB2.

To perform this task you need to have the TWSWEBUIAdministrator role.

You must have switched the Dynamic Workload Console settings repository from a local file to a database repository, as described in Changing settings repository.

Only users with database administrator rights are authorized to initialize the Dynamic Workload Console related tables on the database.

If you want the Dynamic Workload Console to access the database repository with a user without database administrator privileges you must follow these steps:

1. Create a new DB2 user and grant this user with SELECT, INSERT, UPDATE, DELETE rights on all the following tables, belonging to TDWC schema:

```
TDWC_EngineConnection
TDWC_QueryTask
TDWC_ReportTask
TDWC_MEQueryTask
TDWC_Credential
TDWC_ConfigurationProperty
TDWC_Preferenceable
```

The above are the default permissions. However, if you need to restrict your policy, you can give the following permissions to the new DB2user:

```
revoke connect,bindadd, createtab, implicit_schema on database from public;
revoke use of tablespace USERSPACE1      from public;
```

```
grant use of tablespace userspace1 to user twsdb2;
grant createtab on database to user twsdb2;
grant implicit_schema on database to user twsdb2;
```

2. Change Dynamic Workload Console user accessing DB2



- a. From the navigation toolbar, click **System Configuration > Manage Settings**.

- b. In the **Database Settings** section, specify the credentials of the newly created user that must to connect to the database.

Note: As a result of this user switch, the Dynamic Workload Console without database administrator privileges will no longer be authorized to the following actions in the Manage Settings panel:

- **Initialize database**
- **Import settings with Cancel and re-create option.**

Chapter 6. Configuring High Availability

How to configure, change, and share your settings repository.

Performance can be highly improved by migrating the Dynamic Workload Console to High Availability Configuration, so as to have multiple console instances working at the same time as one console.

If you use a Dynamic Workload Console in High Availability configuration, when you connect to a Dynamic Workload Console you are not actually connecting to a specific console but to a load balancer that dispatches and redirects the connections among the nodes in the configuration. Therefore, for example, if a node fails, new user sessions are directed to other active nodes in the configuration and this change is completely transparent to users.

To implement this kind of configuration, you must perform the following steps:

1. Change your settings repository as described in *Changing settings repository*.
2. Perform and verify the High Availability configuration for the Dynamic Workload Console as described in *Administration guide > Configuring the Dynamic Workload Console > Configuring High Availability for Dynamic Workload Console*

Changing settings repository

Changing the settings repository.

- To perform this task you need to have the TWSWEBUIAdministrator role.
- You need to have access to an installed DB2 where a database has already been created. If you need information about how to create a DB2 database, see IBM DB2 Database for Linux, UNIX, and Windows Information Center.
- You must have Administrator rights on that database.

User settings such as user preferences, saved tasks, and engine connections are stored in the settings repository, which by default is a local file. However, you can decide to have your settings repository on a database for all Dynamic Workload Console operations that involve user settings.

This can be useful, for example, for scalability purposes or to have multiple Dynamic Workload Console instances sharing the same user settings.

To use a database for your settings repository, you must configure the database settings, as described in the following procedure:

1. Run the wastool:
 - a. From the Tivoli Workload Scheduler installation path, open the `install_dir\wastools` directory and run `installTDWCDataSource wastool` to create the data source.
 - b. Specify the connection details of the DB2 database in `TDWCDataSource.properties` file, located in `install_dir\wastools` directory.
2. Restart the Dynamic Workload Console.
3. Export your settings:



- a. From the navigation toolbar, click **System Configuration > Manage Settings**.
 - b. Optionally, in the Manage Settings panel, click **Export Settings** and save the XML file in a directory of your choice. In this way you save your user settings in a local file to load them on the database, when it becomes your settings repository.
4. Switch repository to DB2.
 - a. In the same panel, click **Configure settings repository > Use database as settings repository** to specify that settings must be saved in the database instead of a local file.
 - b. In the **Database connection properties** section, specify the credentials required to connect to the database.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.

 - c. Optionally, you can test the connection.
 - d. Save the new configuration.
 5. Import your settings or initialize the database:
 - a. Optionally, click **Import settings** to import your user settings from the XML file to the database repository. During the import operation, keep the default choice, which overwrites the existing settings with the new settings. Performing this step, the database is automatically initialized.
 - b. If you have not performed previous step, click **Configure settings repository > Initialize database**.

As a result, all your existing user settings relating to current Dynamic Workload Console are saved in the database, and all the operations involving user settings are run using the settings in this repository.

Chapter 7. Customizing your console

How to customize your Console.

When you log in to the Dashboard Application Services Hub, you see a navigation bar on the left side with an entry for each Tivoli product installed inside it, such as, for example, the Dynamic Workload Console.

By clicking the hyperlinks displayed in the Dashboard Application Services Hub welcome page, you can learn how to customize your pages, manage roles and perform the most common console administrative tasks.

With respect to the Dynamic Workload Console, in the following sections you can see how to customize the list of your tasks and the startup page to include only the entries and the pages that you really need to access. In addition to this, you can see how to customize the tasks that you need to run and how to enable or disable the notification about product updates.

Customizing your portfolio

How to customize your portfolio.

Several products might be integrated in this portal and their related entries are listed together with the ones belonging to the Dynamic Workload Console in the toolbar displayed on the left side.

You can create a list of your favorite pages, including only the tasks you use most often.



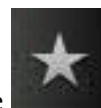
The **Favorites** icon in the welcome page, lets you access your daily tasks. Entries previously displayed in the **My Tasks** list have been automatically added to **Favorites**.

To add a task to Favorites, simply drag it to the Favorites icon. To remove an item from your list of favorites, click and drag the item that you want to remove away from the list.

Alternatively, to add new tasks or manage existing tasks contained in your Favorites list, perform the following procedure:



1. From the welcome page, click the **User > Favorites**.
2. Select only the pages that you want to include among your favorites and click **Apply**.



Your customized list is displayed in the list.

You can also define which pages must be automatically launched when logging into the Dashboard Application Services Hub by adding them to your startup pages.

For additional information about the Dashboard Application Services Hub console customization, refer Dashboard Application Services Hub **Help** section **Customizing the console**.

Customizing your startup page

How to customize the startup page

In the Dashboard Application Services Hub, you can define the list of pages that are launched every time you log in to the console. Use **My Startup Pages** to add or remove dashboards from the list of dashboards or pages that are launched every time you log into the console. Only single task and multiple task dashboards can be added to the startup list. You can set which dashboard is to be opened as the default tab (in focus). To open one of your startup dashboards, click the link for that page in the Page name column.

To add a page to your startup pages, perform the following steps.

1. Open the page you want to be launched when you log in to the console.



2. Click the Page Actions icon at the top right of the panel and select **Add to My Startup Pages**.



3. To remove a page, click **User > My Startup Pages**, select the page in the list and click **Remove**. To specify the page that will be displayed at login time, select **Default**.

The page that you added is launched every time you log in to the console.



To manage the pages you added, click **User > My Startup Pages** in the toolbar on the left. From this page you can remove the pages from the list or define the default page to be displayed when you log in.

For additional information about the Dashboard Application Services Hub, click **Help** at the top right hand corner of the panel.

Customizing your tasks

To customize a task, you must be the task owner.


Configured tasks are provided with default values. However, you can customize their properties and save the newly-modified tasks. You can modify the properties, such as filters and columns, of all the tasks that you own. Starting from a task, you can also customize the properties of secondary queries that are launched starting from the objects resulting from the task.

If you have administrator's rights, you can configure a global setting so as to limit the number of results retrieved by monitor tasks. You can specify the maximum number of items that must be retrieved by the queries, and this configuration will apply to *all* monitor tasks run by current Dynamic Workload Console, except for Monitor Critical jobs. For more information, see “Limit the number of objects retrieved by queries” on page 190.

To customize your tasks, perform the following steps:



1. Click **System Status and Health > All Configured Tasks**.
2. In the All Configured Tasks panel, select a task and click **Task Properties**. The task opens displaying some tabs on the left.

Note: You can access the same task window by running the task and clicking the task properties icon  from the navigation bar, above the table of results.

3. Click the tabs that refer to the properties you want to customize.
4. Modify the values as required and click **Save** to save the modified task.

You have personalized your task. Consider that any customization only applies to the single task you edited. For example, if you change the columns to be displayed in a task named **All Jobs in plan**, this change does not affect any other task to monitor jobs.

Customizing secondary queries


While editing or creating a task, you can also choose the columns to display in *secondary* queries. Secondary queries are those that are run from the table of results of a task. For example, from the list of jobs resulting from a task named **My Jobs**, you can run a secondary query to list all job streams and workstations associated to one of the listed jobs.

You can also customize these secondary queries from the Columns Definition panel of any task. In the Columns Definition panel of **My Jobs** task, you can also choose the columns to display in job stream and workstation tasks. However, this column selection only applies to the lists of job streams and workstations obtained by drilling down from the results of the **My Jobs** task; it does not apply to any other generic monitoring task about job streams or workstations.

Adding tasks to your favorite bookmarks

How to save a task to the favorite bookmarks of your browser.

When you run a task, you can save it as one of the favorite bookmarks of your browser so that you can launch it directly from the browser.

To add a task to your favorite bookmarks, from the panel displaying your task results, click the add link icon .

Related concepts:

“Monitoring your Scheduling Environment” on page 127

“Monitoring your Workload” on page 130

Using the news notification beacon

How to be always up to date with the latest news and information about the product and receive internal communications from the environment administrator.

A beacon appears on your screen when an update for the product or an internal communication from the administrator is made available. Click the beacon to open a pop-up that describes the update and gives you a direct link to it.

Update and news notifications relate to different topics, belonging to categories, such as:

- APARs
- Fixes and utilities
- News
- Technotes
- Product documentation and publications
- Internal communications

By default, the news notification is enabled for all users to all the categories, however, optionally, the TWSWEBUIAdministrator can customize this behavior specifying which user roles must receive all the notifications or only some of them or even none. For more information, see: “Disable and customize NewsFeed function” on page 185.

The news notification is enabled by default, however, to unsubscribe from all the news, or only from the displayed news category, you can select the related check-box in the news pop-up.

In the pop-up window that displays the latest news, you can also navigate through past news and notifications and provide your feedback on a piece of news through an IBM feedback form.

If you have TWSWEBUIAdministrator role or writing rights on the system where Dynamic Workload Console, is installed, you can add multiple **NewsFeed** sections in the `TdwcGlobalSettings.xml` file to transmit different notifications to different user roles. For more information, see “Sending internal communications” on page 26.

Disabling news notification

How to enable and disable news notification preventing the beacon from appearing on your screen when an update for the product is made available.

By default, all users receive all the news notifications. However, optionally, you can customize this behavior by preventing all or only some specific news from being notified to you.

When the beacon appears on your screen and you click it, the first notification displays in a pop-up. If you do not want to receive any notifications at all, or if you want to unsubscribe from some specific categories of notifications, select the related checkbox in the news pop-up.

To enable again all the news notifications, perform the following steps:



1. From the navigation toolbar, click **System Configuration > Set User Preferences**. The Set User Preferences panel containing the current settings is displayed.
2. Click **Restore beacon defaults**.

Related reference:

“Disable and customize NewsFeed function” on page 185

Customizing the list of notification recipients

How to customize the list of recipients of the news notification beacon, by specifying who must receive what kind of notification or communication.

You must have TWSWEBUIAdministrator role or writing rights on the system where Dynamic Workload Console is installed.

By default, the news notification is enabled for all users to all the categories, however, optionally, the TWSWEBUIAdministrator can customize this behavior specifying which user roles must receive all the notifications and internal communications, or only some of them or even none. You can send multiple feeds, containing different communications, addressed to different recipients by customizing the `TdwcGlobalSettings.xml` file and providing multiple feeds, as explained in “Sending internal communications” on page 26.

For example, often operators do not need to have information about new patches and fix packs to be installed or about product events. Analysts are only interested in information and updates about reporting. Whereas, Administrators want to receive all notifications (and, optionally, force the default settings and send a notification to those who are normally excluded from a category).

Customize the `TdwcGlobalSettings.xml` file to specify the news categories that must be notified to users, based on their roles.

A template of this file is located on the installation DVD under `/utilities/TdwcGlobalSettings.xml`. You can modify it, replacing default values with customized ones and enabling commented sections. After customizing, you must copy it under the following path: *Installation_dir/profile_dir/registry* directory, where:

The `TdwcGlobalSettings.xml` file is organized into several sections that can be repeated multiple times in the same file and applied differently to different user roles. Therefore, to customize a behavior based on the user role, uncomment the section corresponding to the user role and include the sections that must apply to this role.

The **Disable and customize NewsFeed function** section contains the configuration details regarding the notification beacon. Therefore, for example, to enable the notifications of specific categories only to a specific user role, you can insert a section like the following:

```
<settings role="TWSWEBUIAdministrator">
<NewsFeed>
<property name="NewsFeed" type="RSS" value="http://www.my.company.com/
RSS_administrators.xml" />
<property name="NewsFeedCategory" value="Administrators"
icon="http://www.my.company.com/
administrator.png" />
</NewsFeed>
</settings>
<settings role="TWSWEBUIOperator">
```

```

<NewsFeed>
<property name="NewsFeed" type="RSS" value="http://www.my.company.com
/RSS_operators.xml" />
<property name="NewsFeedCategory" value="Operators"
icon="http://www.my.company.com/operators.png" />
</NewsFeed>
</settings>

```

If no customized feed is specified, the default feed is used, which retrieves the latest product information from official support sites. To disable any notification, comment the entire section. To disable only external notifications about product information updates, assign an empty string as value to the FeedURL property of JSONP feed like:

```
<property name="FeedURL" type="JSONP" value="" />
```

Related reference:

“Disable and customize NewsFeed function” on page 185

Sending internal communications

How to use the news feed beacon to transmit internal communications to all or some Dynamic Workload Console users.

You must have TWSWEBUIAdministrator role or writing rights on the system where Dynamic Workload Console, is installed.

You can use the news notification beacon to send specific communications to internal Dynamic Workload Console users, simply by storing these communications on the Dynamic Workload Console server. For example, administrators can use this feature to broadcast maintenance notices or company related announcements.

Note:

If no customized feed is specified, the default feed is used, which retrieves the latest product information from official support sites. To disable any notification, comment the entire section. To disable only external notifications about product information updates, assign an empty string as value to the FeedURL property of JSONP feed like:

```
<property name="FeedURL" type="JSONP" value="" />
```

To add customized feeds, specify a list of sections named **NewsFeed** in the TdwcGlobalSettings.xml file by performing the following procedure:

1. Edit the TdwcGlobalSettings.xml file by adding the properties specifying the name and format of the file in the **NewsFeed** section. **Example:**

```

<property name="NewsFeed" type="RSS"
value="http://www.DWC_hostname:portnumber.com/news_rss.xml"/>

```

For more details, see the “Customizing your global settings” on page 27.

2. Optionally, specify a polling interval for the feed reading and an image associated to the notification. **Example:**

```

<property name="PollInterval" value="600" />
<property name="PollInitialDelay" value="1" />
<property name="NewsFeedCategory" value="my company info"
icon="http://www.my.company.com/info.png" />
<property name="NewsFeedCategory" value="my company alert"
icon="http://www.my.company.com/alert.png" />

```

Note: To specify multiple feeds, you must specify multiple **NewsFeed** properties.

3. Optionally, specify a list of feeds that must be taken into consideration by the notification feature. The browser detects new items from all the specified feeds and merge them sorting them by date in a single list.
4. Write the communication in ATOM 1.0 or RSS 2.0 format and store this file in the an HTTP server complying with the *same origin policy*. For browser security reasons, this policy permits to access information only on server using the same protocol, hostname and port number as the one to which you are connected.
5. Optionally, if you want to store your customized feed on an external server, you must configure an HTTP reverse proxy server mapping the external server address.

The notification beacon function reads all the feeds specified in the **NewsFeed** section, detects new information and aggregates all in a single file ordering it by date. Then, it transmits notifications and internal communications to the recipients based on the roles specified in the `TdwcGlobalSettings.xml` file.

Related reference:

“Disable and customize NewsFeed function” on page 185

Customizing your global settings

Some general settings of the Dynamic Workload Console can be included in a customizable file named `TdwcGlobalSettings.xml`. A copy of this file is installed locally in the following path after you install the Dynamic Workload Console:

`<JazzSM_profile_dir>/profile/registry`

You can also find a copy of this file, in template form, on the installation DVD in the directory `/utilities/TdwcGlobalSettings.xml`.

Modify the file replacing default values with customized ones and enabling commented sections, and then save the file to the `<JazzSM_profile_dir>/registry` directory. The complete path of the file is as follows:

For Windows:

`C:\Program Files\IBM\JazzSM\profile\registry\TdwcGlobalSettings.xml`

For UNIX/Linux:

`/opt/ibm/JazzSM/profile/registry/TdwcGlobalSettings.xml`

Users with Administrator privileges can use a configuration file, named `TdwcGlobalSettings.xml`, to add and modify some customizable information, such as:

- The maximum number of objects shown in the graphical views.
- The setting to display the plan view in a new window.
- The configuration details to enable the news notification beacon and be constantly up-to-date with product information. See Disabling news notification.
- The creation of predefined tasks.
- The URLs where you can store customized documentation about your jobs or job streams to associate customized documentation to them.
- The current user registry in use.
- The timeout to read and write information on a Tivoli Workload Scheduler for z/OS engine.

- The maximum number of objects retrieved with a query, the maximum number of rows to display in a table, and the maximum number of direct queries to maintain in history.
- Allowing or preventing users from sharing tasks and engine connections.
- The display of all dependencies, both satisfied and unsatisfied.
- The use of audit files to track activities in the Self-Service Catalog and Self-Service Dashboards mobile applications.

A template of this file is located on the installation DVD under `/utilities/TdwcGlobalSettings.xml`. You can modify it, replacing default values with customized ones and enabling commented sections. After customizing, you must copy it under the following path: *Installation_dir/profile_dir/registry* directory, where:

profile_dir

is the directory you have defined as your profile directory. By default it is `JazzSM/profile`.

For example, the complete path to this file is as follows:

For Windows:

`C:\Program Files\IBM\JazzSM\profile\registry\TdwcGlobalSettings.xml`

For UNIX/Linux:

`/opt/ibm/JazzSM/profile/registry/TdwcGlobalSettings.xml`

This file is accessed at each login, and all configurations specified in the file are immediately applied, except for **precannedTaskCreation** property. This property is read only when a user logs in for the first time and then is used whenever this user logs in again.

You can use any text or XML editor to edit this file, but ensure that you save it is as a valid XML file.

The file is organized into the following sections that group similar properties:

Sections can also be repeated multiple times in the same file and applied differently to different user roles. To apply a section only to the users belonging to a role, the section must be included within the following setting:

settings role

The user for which the following configuration must be applied. Default value: all users, unless otherwise specified.

Only one **settings** section can be specified for each role. If a user has more than one role, the settings associated to the higher role are taken into consideration.

Example:

```
<?xml version="1.0"?>
<tdwc>
.
.
<settings>
<graphViews>
<property name="planViewNewWindow" value="true"/>
</graphViews>
</settings>

<settings role="TWSWEBUIOperator">
```



```
<graphViews>
<property name="planViewNewWindow" value="false"/>
</graphViews>
</settings>
.
.
</tdwc>
```

See “TdwcGlobalSettings.xml sample” on page 194 to view the complete syntax for the file.

Chapter 8. Tivoli Workload Scheduler Concepts

Conceptual information about Tivoli Workload Scheduler

This section provides conceptual information about Tivoli Workload Scheduler and the Dynamic Workload Console.

Scheduling environment

This section contains the main concepts that help you to understand what a scheduling environment is and what it comprises.

Workstation

Note: This section provides information relating to the use of workstations for scheduling jobs and job streams.

If, instead, you want to learn about workstations because you are planning your network, see Planning and Installation or Planning and Installation

The computer system where you run your jobs and job streams is called a *workstation*. Workstations can be grouped logically into *workstation classes* and organized hierarchically into *domains*, managed by *domain managers*.

When you create a workstation definition for a system in your network you define a set of characteristics that uniquely identify the system and that control how jobs run on it. For example, the IP address of the workstation, if it is behind a firewall, if communications with it must be secure, what time zone it is in, and the identity of its domain manager.

Workstations in the Tivoli Workload Scheduler scheduling network perform job and job stream processing, but can also have other roles. When your network is designed, these roles are assigned to these workstations to suit the specific needs of your business. The following types of workstation are available:

Distributed **Master domain manager**

A workstation acting as the management hub for the network. It manages all your scheduling objects. The master domain manager workstation must be installed with this role.

Distributed **Backup master domain manager**

A workstation that can act as a backup for the master domain manager when problems occur. It is a master domain manager, waiting to be activated. Its use is optional. This workstation must be installed as a master domain manager workstation.

Learn more about switching to a backup master domain manager in Administration Guide.

Distributed **Domain manager**

A workstation that controls a domain and that shares management responsibilities for part of the Tivoli Workload Scheduler network. It is installed as an agent, and then configured as a domain manager workstation when you define the workstation in the database.

Dynamic domain manager

An installed component in a distributed Tivoli Workload Scheduler network that is the management hub in a domain. All communication to and from the agents in the domain is routed through the dynamic domain manager. When you install a dynamic domain manager the workstation types listed below are created in the database:

fta Fault-tolerant agent component manually configured as domain manager

broker Broker server component

agent Dynamic agent component

Backup dynamic domain manager

A workstation which can act as a backup for the dynamic domain manager, when problems occur. It is effectively a dynamic domain manager, waiting to be activated. Its use is optional.

Learn more about switching to a backup dynamic domain manager in Administration Guide. When you install a dynamic domain manager the workstation types listed below are created in the database:

fta Fault-tolerant agent component.

broker Broker server component

agent Dynamic agent component

Fault-tolerant agent

A workstation that receives and runs jobs. If there are communication problems with its domain manager, it can run jobs locally. It is installed as an agent, and then configured as a fault-tolerant agent workstation when you define the workstation in the database. This workstation is recorded in the Tivoli Workload Scheduler database as **fta**.

Standard agent

A workstation that receives and runs jobs only under the control of its domain manager. It is installed as an agent, and then configured as a standard agent workstation when you define the workstation in the database.

Extended agent

A workstation that has a host and an access method. The host is any other workstation, except another extended agent. The access method is an IBM-supplied or user-supplied script or program that is run by the host whenever the extended agent is referenced in the production plan. Extended agents are used to extend the job scheduling functions of Tivoli Workload Scheduler to other systems and applications. For example, to launch a job on an extended agent, the host runs the access method, passing it job details as command line options. The access method communicates with the external system or application to launch the job and return the status of the job.

Also it is a workstation where a Tivoli Workload Scheduler for Applications access method has been installed as a bridge so that you can schedule jobs in the SAP R/3, Oracle E-Business Suite, PeopleSoft, z/OS, or custom applications. It must be physically hosted by a fault-tolerant agent (up to 255 extended agents per fault-tolerant agent) and then defined as an extended agent in the database.

For more information, see User's Guide and Reference and Tivoli Workload Scheduler for Applications User's Guide

Workload broker agent

A workstation that manages the lifecycle of Tivoli Workload Scheduler Workload Broker jobs in Workload Broker. It is installed and configured as a dynamic workload broker workstation in the database.

TWS for z/OS agent

A distributed workstation that runs jobs scheduled from Tivoli Workload Scheduler for z/OS. Like fault-tolerant workstations, it is installed in a Tivoli Workload Scheduler distributed domain. Unlike fault-tolerant workstations, it does not:

- Have fault tolerance
- Require an end-to-end server
- Need topology definitions

Communication with the agents is handled directly by the controller. For more information about the end-to-end scheduling with fault tolerance capabilities, see Tivoli Workload Scheduler for z/OS Scheduling End-to-end with z-centric Capabilities.

z/OS

Virtual workstation

A workstation that is created with the automatic reporting attribute and the virtual option, defining a list of destinations, for the workload submission, that are used to spread the workload across trackers. When the scheduler processes the jobs submitted to a virtual workstation, it distributes the workload according to a sequenced turn criteria, based on a round-robin algorithm. To submit the job, at least one of the destinations in the list must be available.

You can associate open intervals, parallel servers, and fixed resources to each destination belonging to the defined pool. The association is disabled at virtual workstation level, because the jobs that you submit on a virtual workstation are actually run on a single destination. When you associate parallel servers with a virtual workstation destination, you can specify a value up to 65535. The alternative workstation definition is not applicable either at workstation level or at single destination level.

Remote engine

A workstation that represents locally a remote Tivoli Workload Scheduler engine. It is a workstation used to run only *shadow jobs*. A shadow job is a job that runs locally and is used to map another job running on a remote engine. This relationship between the two jobs is called a *cross dependency*. You define a remote engine workstation if you want to federate your environment with another Tivoli Workload Scheduler environment, either distributed or z/OS, to add and monitor dependencies on jobs running in the other scheduling environment. This type of workstation uses a connection based on HTTP protocol to allow the two environments to communicate.

Dynamic agent

A workstation that manages a wide variety of job types, for example, specific database or FTP jobs, in addition to existing job types. This workstation is automatically created and registered when you install the dynamic agent. Because the installation and registration processes are performed automatically, when you view the agent in the Dynamic Workload Console, it results as updated by the Resource Advisor Agent. You can group agents in pools and dynamic pools.

In a simple configuration, dynamic agents connect directly to a master domain manager or to a dynamic domain manager. However, in more complex network topologies, if the network configuration prevents the master domain manager or the dynamic domain manager from directly communicating with the dynamic agent, then you can configure your dynamic agents to use a local or remote gateway.

Pool A workstation grouping a set of dynamic agents with similar hardware or software characteristics to submit jobs to. Tivoli Workload Scheduler balances the jobs among the dynamic agents within the pool and automatically reassigns jobs to available dynamic agents if an agent is no longer available. To create a pool of dynamic agents in your Tivoli Workload Scheduler environment, define a workstation of type **pool** hosted by the workload broker workstation, then select the dynamic agents you want to add to the pool. A computer system group is automatically defined in the workload broker database together with its associated dynamic agents.

Dynamic pool

A workstation grouping a set of dynamic agents that is dynamically defined based on the resource requirements you specify. For example, if you require a workstation with low CPU usage and the Windows operating system installed to run your job, you specify these requirements using the Dynamic Workload Console or the **composer** command. When you save the set of requirements, a new workstation is automatically created in the Tivoli Workload Scheduler database. This workstation is hosted by the workload broker workstation. This workstation maps all the dynamic agents in your environment that meet the requirements you specified. The resulting pool is dynamically updated whenever a new suitable agent becomes available. Jobs scheduled on this workstation automatically inherit the requirements defined for the workstation.

Related tasks:

"Creating distributed workstations" on page 75

"Creating z/OS workstations" on page 76

"Creating z/OS virtual workstations" on page 77

"Creating a task to Monitor Workstations" on page 127

"Creating Workload Broker objects" on page 107

Related reference:

"Workstation types" on page 206

Domain

Distributed

The domain.

All the workstations in a distributed Tivoli Workload Scheduler network are organized into one or more *domains*, each of which consists of one or more agents and a domain manager acting as the management hub. Most communication to and from the agents in the domain is routed through the domain manager. If the agent has the "behind firewall" designation, all of it is.

All the networks have a master domain where the domain manager is the master domain manager. It maintains the database of all the scheduling objects in the domain and the central configuration files. The master domain manager generates

the plan and creates and distributes the Symphony file. In addition, logs and reports for the network are maintained on the master domain manager.

You can organize all the agents in your network into a single domain or into multiple domains.

Single-domain network

A single domain network consists of a master domain manager and any number of agents. Figure 1 shows an example of a single-domain network. A single-domain network is well suited to companies that have few locations and business functions. All the communication in the network is routed through the master domain manager. With a single location, you are concerned only with the reliability of your local network and the amount of traffic it can handle.

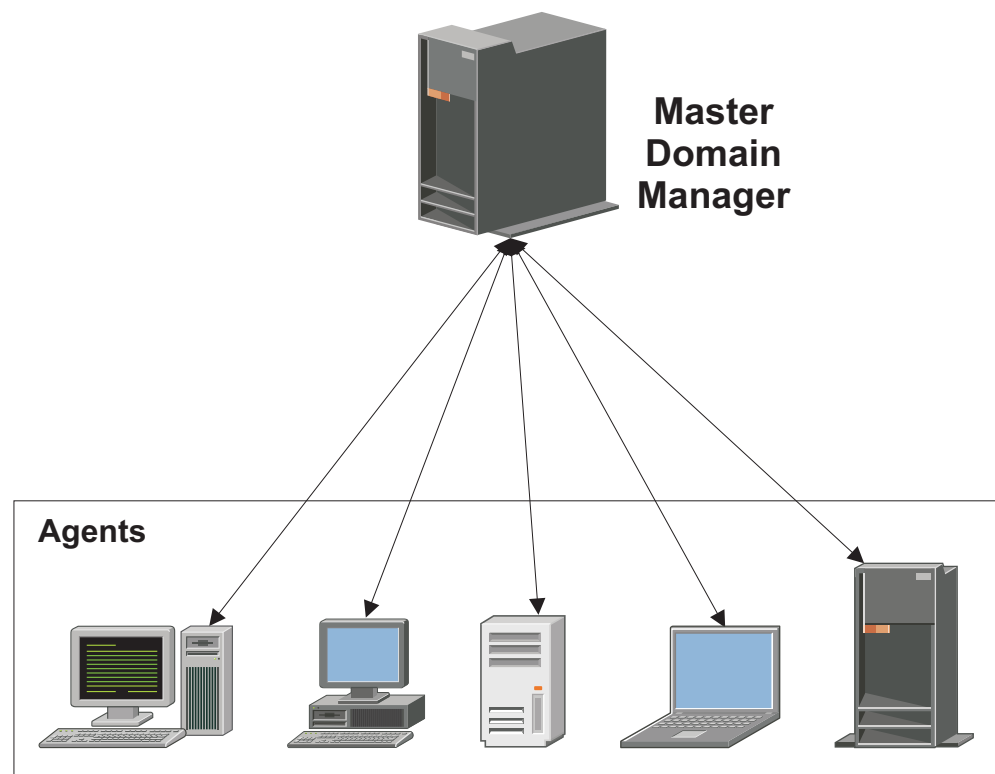


Figure 1. Single-domain network

Multiple-domain network

Multiple-domain networks are especially suited to companies that span multiple locations, departments, or business functions. A multiple-domain network consists of a master domain manager, any number of lower tier domain managers, and any number of agents in each domain. Agents communicate only with their domain managers, and domain managers communicate with their parent domain managers. The hierarchy of domains can have any number of levels.

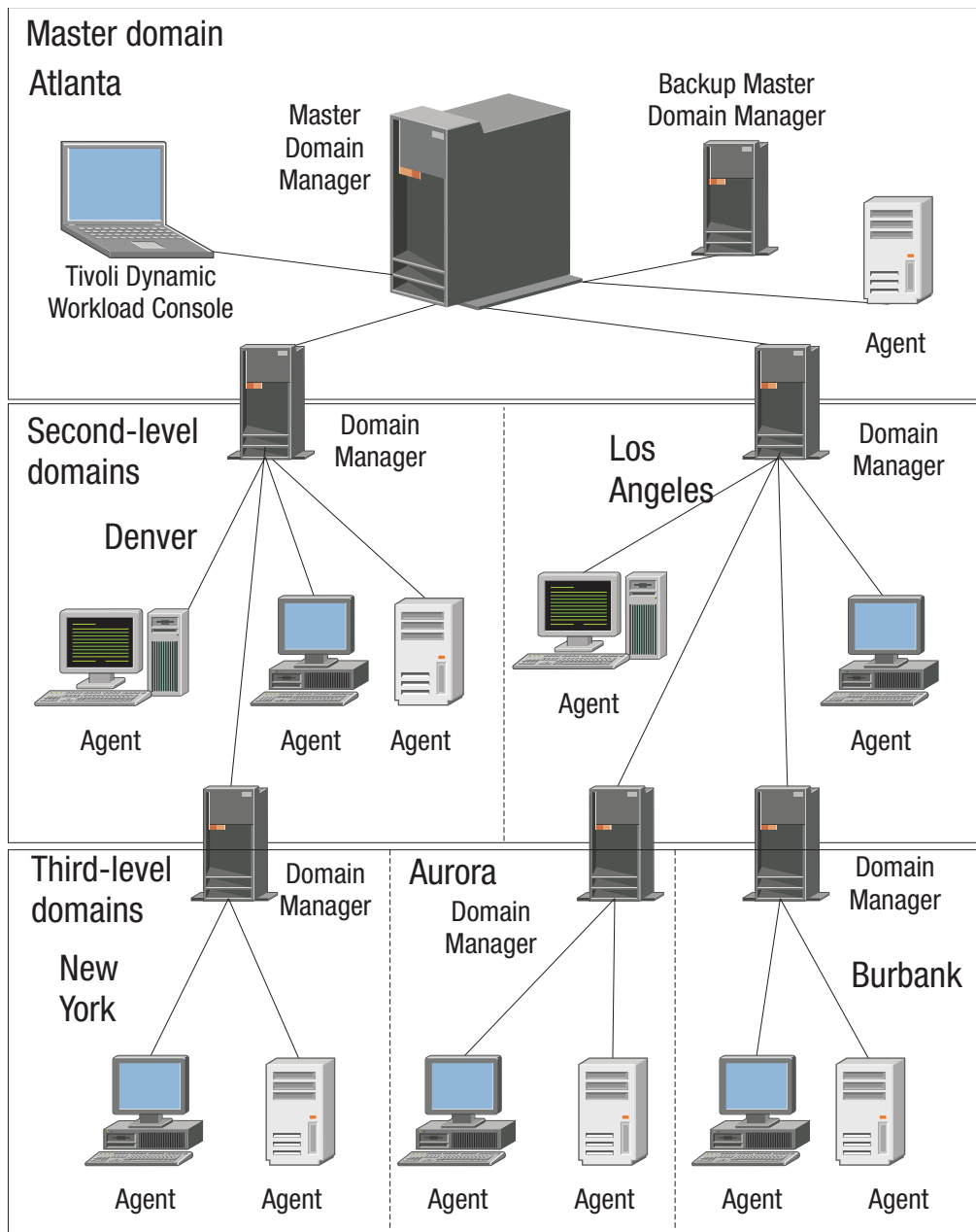


Figure 2. Multiple-domain network

In Figure 2, the master domain manager is located in Atlanta. The master domain manager contains the database files used to document the scheduling objects, and distributes the Symphony file to its agents and to the domain managers in Denver and Los Angeles. The Denver and Los Angeles domain managers then distribute the Symphony file to their agents and subordinate domain managers in New York, Aurora, and Burbank. The master domain manager in Atlanta is responsible for broadcasting inter-domain information throughout the network.

All the communication to and from the New York domain manager is routed through its parent domain manager in Denver. If there are schedules or jobs in the New York domain that are dependent on schedules or jobs in the Aurora domain, those dependencies are resolved

by the Denver domain manager. Most inter-agent dependencies are handled locally by the lower tier domain managers, greatly reducing traffic on the network.

You can change the domain infrastructure dynamically as you develop your network. You move a workstation to a different domain, by changing the domain name in its database definition. The change takes effect when the master generates/extends the plan.

Tip: You cannot schedule jobs or job streams to run on all workstations in a domain by identifying the domain in the job or job stream definition. To achieve this, you must create a *workstation class* that contains all the workstations in the domain.

Related tasks:

“Creating a domain” on page 78

“Creating distributed workstations” on page 75

“Creating a task to Monitor Domains” on page 129

Scheduling objects

The set of scheduling objects described in the current plan is a subset of all the scheduling objects stored in the database. The scheduling objects accessible from the Dynamic Workload Console depend on your Tivoli Workload Scheduler environment.

Distributed For distributed environments, the scheduling objects reported in the production plan are:

- All the active workstations defined in the database. These are the workstations whose definition does not have the *ignore* flag set to on.
- All the domains.
- All the job streams scheduled to start in the production period and all jobs belonging to these job streams.
- All the resources, files, parameters, variables, and prompts defined in the job streams.

z/OS For z/OS environments, the scheduling objects reported in the current plan are:

- All the active workstations defined in the database.
- All the job streams scheduled to start in the production period and all jobs belonging to these job streams.
- All the resources that these jobs and job streams depend on.

To differentiate between jobs and job streams defined in the database and jobs and job streams scheduled to run within the production period, according to the Tivoli Workload Scheduler standard naming convention, each occurrence of a job or a job stream scheduled to run in the current plan is called an *instance*. A current plan can contain more than one instance of the same job or job stream.

Related concepts:

“Designing your Scheduling Environment” on page 75

“Monitoring your Scheduling Environment” on page 127

“Monitoring your Workload” on page 130

Related tasks:

“Creating and managing engine connections” on page 8

“Designing your Workload” on page 79

Job

A *job* is a unit of work that specifies an action, such as a weekly data backup, to be performed on specific workstations in the Tivoli Workload Scheduler network.

Distributed In a Tivoli Workload Scheduler distributed environment, jobs can be defined either independently from job streams or within a job stream definition.

z/OS In a Tivoli Workload Scheduler for z/OS environment, jobs can be defined only within a job stream and are called *operations*. You can have started-task operations, which are operations run on a computer workstation that are used to start and stop started tasks.

Regardless of whether the Tivoli Workload Scheduler engine is distributed or z/OS based, you can define locally a *shadow job* to map a remote job instance running on a different Tivoli Workload Scheduler engine.

Related tasks:

“Creating a task to Monitor Jobs” on page 132

“Creating a task to Monitor Critical Jobs” on page 134

“Creating a task to Monitor Tasks on Multiple Engines” on page 137

“Adding a job to a job stream” on page 94

“Listing jobs and job streams” on page 115

“Creating job stream definitions” on page 83

“Creating job definitions” on page 84

“Prerequisite steps to create job types with advanced options” on page 91

Related reference:

“Status description and mapping for distributed jobs” on page 199

“Status description and mapping for z/OS jobs” on page 201

“Limit the number of objects retrieved by queries” on page 190

Job stream

A *job stream* is a sequence of jobs to be run, together with times, priorities, and other dependencies that determine the order of processing. Each job stream is assigned a time to run, represented by run cycle with type calendar, set of dates, or repetition rates.

z/OS In a Tivoli Workload Scheduler for z/OS environment, job streams are called *applications*.

Related concepts:

“Customizing your job stream” on page 171

Related tasks:

“Creating a task to Monitor Job Streams” on page 138

“Creating a task to Monitor Job Streams on Multiple Engines” on page 140

“Creating job stream definitions” on page 83

“Adding a job to a job stream” on page 94

“Listing jobs and job streams” on page 115

Related reference:

“Status description and mapping for distributed job streams” on page 203

“Status description and mapping for z/OS job streams” on page 205

“Limit the number of objects retrieved by queries” on page 190

Workload application

A workload application is one or more job streams together with all the referenced jobs that can be shared with other Tivoli Workload Scheduler environments through an easy deployment process.

A workload application is a Tivoli Workload Scheduler database object that acts as a container for one or more job streams. You can use workload applications to standardize a workload automation solution so that the solution can be reused in one or more Tivoli Workload Scheduler environments thereby automating business processes.

You can prepare a workload application template in a source Tivoli Workload Scheduler environment and then export it so that it can be deployed in a target environment. The export process extracts from the source environment all of the elements necessary to reproduce the solution in another environment. It produces a compressed file containing a number of files required to import the workload application into the target environment. These files contain a definition of the objects in the source environment extracted from the Tivoli Workload Scheduler database. For those elements that depend on the topology of the target environment, some manual configuration is required. For example, the definitions extracted from the source environment contain references to workstations that do not exist in the target environment. For this reason, before proceeding with the import, a mapping of some of the elements must be made associating the name of the object in the target environment.

The exported workload application template contains definitions or references for all of the following objects:

- Job streams
- Jobs
- Workstations, workstation classes
- Calendars
- Prompts
- Run cycles
- Run cycle groups
- Resources
- Internetwork dependencies
- External dependencies

Period

z/OS

Periods are either *cyclic*, such as a week or 28-day period, or *noncyclic*, such as an academic semester.

Cyclic periods

Defined by their origin date and their length: a cyclic period starts on a specific date and has a specified number of days. There are two kinds of cyclic periods:

work-days-only cyclic periods

Only working days are considered.

all-days cyclic periods

All the days are considered.

Noncyclic periods

Defined by the origin date of each interval and can optionally have an end date for each interval.

Periods can be combined with offsets to create run cycles and define when a job stream runs. For example, an offset of 1 in a weekly period specifies Monday. An offset of 10 in a monthly period specifies the tenth day of each month.

The long-term planning process uses the calendar information, the period definitions, and the run cycle, to determine the days on which an application is scheduled to run.

If you run workload on fixed days of the week, month, or year, and take one of the standard Tivoli Workload Scheduler for z/OS actions when this day falls on a non-working day, you do not need to create your own periods. You can describe most cases with rules such as:

- First Sunday in June
- First working day in the week
- Last Friday in the year
- Last non-working day in the month

If you use rules with their built-in calendar cycles (days of the week, months of the year, and so on), you probably need to create only special noncyclic periods, such as university semesters and tax years. The following sections show some examples of types of periods.

Cyclic period examples

Examples of cyclic periods are a day, a week, and a fortnight, with fixed intervals of 1 day, 7 days, and 14 days, respectively. An academic semester cannot be described as a cyclic period, because spring, summer, and fall semesters have different lengths. The following example shows a lunar calendar month, assumed to be 28 days:

Period name

Moon

Type Cyclic based on all days

Interval

28 days

Interval origin

7 February 2009 (date of a new moon)

Non-cyclic period examples

Examples of non-cyclic periods are a quarter and a payroll period. You specify the start of each interval of a non-cyclic period with an origin date. This example shows a period for university semesters, with the interval origin and end specified for each semester:

Period name

Semester

Type Non-cyclic

Interval origin

26 August 2009, 13 January 2010, 9 June 2010.

Interval end

13 December 2009, 16 May 2010, 28 June 2010

Non-cyclic periods have a once-a-year maintenance overhead when you must create the intervals for the coming months. For this reason, carefully consider how flexible your period definitions are, and remove potentially duplicated definitions.

Calendar

A *calendar* is a list of dates that define when a job stream runs.

Distributed

Calendar in a distributed environment

A calendar can also be designated as a *non-working days* calendar in a job stream. A non-working days calendar is a calendar that is assigned to a job stream to represent the days when the job stream and its jobs do not run. It can also be used to designate Saturdays or Sundays, or both, as workdays. The default non-working days calendar for all job streams is called the *holidays* calendar.

z/OS

Calendar in a z/OS environment

The calendar specifies normal working days and public holidays. Tivoli Workload Scheduler for z/OS uses the calendar to determine when job streams are scheduled and to calculate dates for JCL tailoring.

You can specify the calendar when you create a job stream. If no calendar is specified for the job stream, Tivoli Workload Scheduler for z/OS uses the calendar in the CALENDAR keyword of the BATCHOPT initialization statement, for batch services such as extending the long-term plan, or the calendar specified under the Tivoli Workload Scheduler for z/OS options, for online services such as testing a rule with GENDAYS.

If no calendar is specified, a calendar with the name DEFAULT is used. If the DEFAULT calendar does not exist, all days are considered as working days. You can have several calendars, but always name your default calendar DEFAULT, and specify the same calendar name on BATCHOPT or in the Tivoli Workload Scheduler for z/OS options. A calendar must contain at least one working day.

Related tasks:

“Designing your Workload” on page 79

“Listing workload objects” on page 114

Run cycle

A *run cycle* specifies the days that a job stream is scheduled to run. Each run cycle is defined for a specific job stream and cannot be used by other job streams. You can specify the following types of run cycle:

Distributed simple

A specific set of user-defined days when a job stream is run.

daily A run cycle that specifies that the job stream runs according to a day frequency and type that you set. For example, it might run daily, every three days, or just on working days.

weekly

A run cycle that specifies the days of the week when a job stream is run. For example, a job stream can be run every Monday, Wednesday, and Friday using a weekly run cycle.

monthly

A run cycle that specifies that the job stream runs according to a monthly day or date that you set. For example, it might run every 1st and 2nd day of the month, or every 1st Monday and 2nd Tuesday of the month.

Distributed It can also run, for example, every 1st and 2nd day of the month every two months.

yearly A run cycle that specifies that a job stream runs, for example, yearly.

Distributed It can also run, for example, every three years.

exclusive

A run cycle that specifies the days and times when a job stream cannot be run. Exclusive run cycles take precedence over inclusive run cycles.

inclusive

A run cycle that specifies the days and times when a job stream is scheduled to run. Exclusive run cycles take precedence over inclusive run cycles.

offset-based

A run cycle that uses a combination of user-defined periods and offsets. For example, an offset of 3 in a period of 15 days is the third day from the beginning of the period. It is more practical to use offset-based run cycles when the cycle is based on cyclic periods. This term is used only in Tivoli Workload Scheduler for z/OS, but the concept applies also to the distributed product.

rule-based

A run cycle that uses rules based on lists of ordinal numbers, types of days, and common calendar intervals (or period names in Tivoli Workload Scheduler for z/OS). For example, the last Thursday of every month. Rule-based run cycles are based on conventional periods, such as calendar months, weeks of the year, and days of the week. In Tivoli Workload Scheduler for z/OS, run cycles can also be based on periods that you define, such as a semester. This term is used only in Tivoli Workload Scheduler for z/OS, but the concept applies also to the distributed product. You can also specify a rule to establish when a job stream runs if it falls on a free day.

Related concepts:

“Creating and managing run cycle groups and their run cycles” on page 100

Related information:

“Run Cycle Preview” on page 221

Run cycle group

You can optionally define a run cycle group for your job stream instead of, or in addition to, a number of single run cycles.

A run cycle group is a list of run cycles that are combined together to produce a set of run dates.

By using run cycle groups, you can benefit from the following advantages:

A run cycle group is a distinct database object

It is defined by itself and can be matched with one or more job streams. It is not defined as part of a specific job stream like single run cycles.

The same run cycle group can be used on different job streams

This improves the overall usability of the run cycles, because you can specify the same run cycle group in multiple job streams, avoiding the need to have multiple run cycle definitions for the same scheduling rules.

Run cycle groups enhance the use of exclusive run cycles

Exclusive (or negative) run cycles are used to generate negative occurrences, which identify the days when a job stream would normally be scheduled but is not required. The sum of the exclusive run cycles are subtracted from the inclusive ones. A negative occurrence always cancels any matching positive occurrences and you can specify a negative occurrence only if the positive equivalent already exists. An exact matching of the days, as well as any time restrictions, is required between the exclusive and inclusive run cycles for the cancellation to occur. Run cycle groups add much flexibility by allowing users to apply exclusive run cycles to a subset of the positive ones rather than to all of them. Group your run cycles into *subsets* so that the exclusive run cycles can be applied only to the positive occurrences generated by the run cycles belonging to the same set.

Run cycles must be organized into *subsets* within a run cycle group. The subsets are always in a logical **OR** relationship with each other. The result of the run cycle group is always a date or set of dates; it cannot be negative.

For example, you might want your job stream to run every day of the month except the last day of the month. But, you also want it to be scheduled on the last day of the year (the last day of December). You can define a run cycle group using subsets, as follows:

Subset 1

- **Run cycle 1** - Inclusive run cycle every day of the month
- **Run cycle 2** - Exclusive run cycle on the last day of the month

Subset 2

- **Run cycle 3** - Inclusive run cycle on December 31st

where, run cycle 2 cancels the last day of each month in Subset 1, while run cycle 3 generates December 31st as a separate date and therefore you can schedule the job stream on Dec 31st.

Run cycle groups allow the use of a logical AND between individual run cycles in the subset

By default, the run cycles within a subset are in a logical **OR** relationship but you can change this to a logical **AND**, if the run cycle group result is a

positive date or set of dates (Inclusive). For each run cycle, you can specify either operator (**AND** ,**OR**), obtaining the following behavior:

1. All the run cycles of the group that are in *AND* relationship are calculated first. The result of this calculation is a date or a set of dates.
2. Then, all the run cycles in an *OR* relationship are added to the result of the previous step.

A similar behavior is applied to inclusive and exclusive run cycles to determine the final date or set of dates of a group.

Inclusive (A)

Rule-based run cycle. Select days when the job stream is to be run if they belong to all A types of the set of run cycles.

Exclusive (D)

Exclusion rule-based run cycle. Select days when the job stream is NOT to be run if they belong to all D types of the set of run cycles.

For example, you can add two conditions together:

Run on Wednesday "AND" the 8th workday of the month.

In this way, the only scheduled dates are any 8th work day of the month that falls on a Wednesday.

Full compatibility with *traditional* run cycles

The *traditional* run cycles specified in the job stream definition can reference run cycle groups, with the possibility to specify shift or offsets on them (as with periods for z/OS or calendars for distributed systems).

A set of dates (interval starts) is created automatically either at run cycle level directly (inclusively or exclusively with offsets, or in the rule. This is a two-step process with run cycles:

1. Define the key "business event", such as, Month End, using run cycles and free day rules
2. Define rules that use the dates of the "business event" as the intervals against which the other batch run can be scheduled relative to.

For example, you have a *Month End process* that runs on the Last Friday of a month, but that moves forward to the next working day, except in December when it runs on the 3rd Friday of the month. This scheduling rule can be defined with a few rules, run cycles, and free day rules.

Two working days before Month End you need to run a pre-validation process to allow problems to be addressed before the run. You cannot choose the last Wednesday of the month, because in some months this might occur after the last Friday. Similarly, if the last Friday was a free day, the last Wednesday will not be 2 working days before it, because the Free Day rule applies ONLY to the day the rule falls on, it cannot look at anything else.

Many other batch runs might also need to be run on a certain number of days before or after the Month End, but the same restrictions apply.

You can now define work to run relative to something defined by a combination of run cycles and free day rules.

Use of calendars with run cycles within a run cycle group

Optionally, you can specify more than one calendar to calculate the working and non-working days definition for a run cycle. The primary

calendar is used to calculate which working days are valid, and a secondary calendar is used to calculate specific non-working dates. If the dates calculated according to the secondary calendar match with the working days in the primary calendar, the job is scheduled; if they do not match, the job is not scheduled.

For example, a global company that runs workload in the United States for many other countries needs many calendar combinations to ensure that the batch jobs only run on a day that is a working day both in the United States and the other country. The calendar can be defined at job stream level and, if not specified, a default calendar is used. However, the calendar at run cycle level, whenever defined, can be used as secondary calendar and the job stream (or default) calendar can be used as the primary calendar.

For example, Primary calendar can be *WORKDAYS*, which is defined as MONDAY to FRIDAY excluding US holiday dates. You might also need to calculate the job runs based on calendar *HKWORK*, which is defined as Monday to Friday excluding Hong Kong holiday dates. The job might have several schedules:

- Run on working days, but not the last working day and not Mondays
- Run on Mondays, but not on the last working day
- Run on the last working day

Because each schedule is calculated against the *WORKHK* calendar it is also checked against the *WORKDAYS* calendar to ensure that it is scheduled on a US working day.

The use of time restrictions with run cycle groups

You can specify time constraints to define the time when processing must start or the time after which processing must no longer start. To do this, you can associate *time restrictions* to job, job streams, run cycles, and run cycle groups. When you define a time restriction, you basically obtain a *time*. Because you can associate time restrictions to multiple objects, the following hierarchy shows the order by which the different time restrictions are taken into consideration to actually define when to start the processing:

1. Time restriction defined in the run cycle into the job stream
2. Time restriction defined in the job stream
3. Time restriction defined in the run cycle contained in the run cycle group associated to the job stream.
4. Time restriction defined in the run cycle group associated to the job stream.
5. Start of Day

This means that:

Time restrictions in the job stream

Override and take precedence *over any other time restrictions* defined in the run cycles or run cycle groups associated to the job stream.

No time restrictions in the job stream nor in the run cycle group

The group generates only a date that is the *Start Of Day*. If offsets and free day rules are to be calculated, the calculation always starts from the *Start Of Day*.

Time restrictions in the run cycle group (not in the job stream)

Time restrictions (and possible offset) are calculated starting from the *Start Of Day* and the resulting date and time indicate the start of processing.

Examples

Table 4. Scenario 1. No time restriction in the run cycle group

Run cycle group	Scheduled date	Earliest Start
Run cycle group	10/24	10/24
Run cycle group with offset (+ 3 days)	10/27 (Saturday)	10/27/ (Saturday)
Run cycle group with free day rule	10/29/ (Monday)	0/29/ (Monday)
Run cycle in the job stream with time restrictions		
Run cycle in the job stream with + 4 working days shift	11/02 (Friday)	11/02 (Friday)
Run cycle in the job stream with free day rule	11/02 (Friday)	11/02 (Friday)
Run cycle in the job stream with earliest start +1 1pm	11/02 (Friday)	11/03 (Saturday) 1pm
Run cycle in the job stream without time restrictions		
Run cycle in the job stream with + 4 working days shift	11/02 (Friday)	11/02 (Friday) Start of Day
Run cycle in the job stream with free day rule	11/02 (Friday)	11/02 (Friday) Start of Day

Table 5. Scenario 2. Time restriction in the run cycle group without offset

Run cycle group	Scheduled date	Earliest Start
Run cycle group	10/24	10/24
Run cycle group with calendar offset (+ 3 days)	10/27/ (Saturday)	10/27/ (Saturday)
Run cycle group with free day rule	10/29/ (Monday)	0/29/ (Monday)
Run cycle in the job stream with time restrictions		
Run cycle in the job stream with + 4 working days shift	11/02 (Friday)	11/02 (Friday)
Run cycle in the job stream with free day rule	11/02 (Friday)	11/02 (Friday)
Run cycle in the job stream with earliest start +1 1pm	11/02 (Friday)	11/03 (Saturday) 1pm
Run cycle in the job stream without time restrictions		
Run cycle in the job stream with + 4 working days shift	11/02 (Friday)	11/02 (Friday) Start of Day

Table 5. Scenario 2. Time restriction in the run cycle group without offset (continued)

Run cycle group	Scheduled date	Earliest Start
Run cycle in the job stream with free day rule	11/02 (Friday)	11/02 (Friday) Start of Day

Table 6. Scenario 3. Time restriction in the run cycle group with offset (+1 12:00)

Run cycle group	Scheduled date	Earliest Start
Run cycle group	10/24	10/24
Run cycle group with calendar offset (+ 3 days)	10/27/ (Saturday)	10/27/ (Saturday)
Run cycle group with free day rule	10/29/ (Monday)	10/29/ (Monday)
Run cycle group with offset +1 12:00	10/29/ (Monday)	10/30 12:00 (Tuesday)
Run cycle in the job stream with time restrictions		
Run cycle in the job stream with + 4 working days shift	11/02 (Friday)	11/02 (Friday)
Run cycle in the job stream with free day rule	11/02 (Friday)	11/02 (Friday)
Run cycle in the job stream with earliest start +1 1pm	11/02 (Friday)	11/03 (Saturday) 1pm
Run cycle in the job stream without time restrictions		
Run cycle in the job stream with + 4 working days shift	11/02 (Friday)	11/03 12:00 (Saturday)
Run cycle in the job stream with free day rule	11/02 (Friday)	11/03 12:00 (Saturday)

z/OS Availability of the GENDAYS command at run cycle group level

Using GENDAYS, you can check the result of the combination of all the run cycles in the group.

Related concepts:

“Creating and managing run cycle groups and their run cycles” on page 100

Related information:

“Run Cycle Preview” on page 221

Operator instructions

z/OS

In a Tivoli Workload Scheduler for z/OS environment, some jobs might require specific instructions about how they are to be handled. These instructions are called operator instructions.

An *operator instruction* can be permanent or temporary. A temporary instruction has a validity period associated with it, which specifies when the instruction is valid.

Parameter

A *parameter* is an object to which you assign different values to be substituted in jobs and job streams, either from values in the database or at run time.

Note: You cannot use parameters with extended agent jobs.

Parameters are useful when you have values that change depending on your job or job stream. Job and job stream definitions that use parameters are updated automatically with the value at the start of the production cycle.

Use parameters as substitutes for repetitive values when defining jobs and job streams. For example, using parameters for user logon and script file names in job definitions and for file and prompt dependencies allows the use of values that can be maintained centrally in the database on the master.

Dependencies

Controlling processing using dependencies

When defining job streams and managing the workload in the plan, you can control processing using dependencies.

You can specify the following types of dependencies:

Distributed Dependencies in a distributed environment:

You can have dependencies between jobs, between job streams, or between jobs and job streams. They can be:

Internal dependencies

These are dependencies established between jobs belonging to the same job stream.

External dependencies

These are dependencies between job streams, between job streams and jobs belonging to other job streams, or between jobs belonging to different job streams. The following resolution criteria are used to satisfy these dependencies:

Closest preceding

The closest preceding in time before the instance that includes the dependency.

Same scheduled date

The instance planned to run on the same day.

Within a relative interval

The closest preceding instance within the relative time interval you chose, or, if none is found, the closest following instance within the relative time interval you chose.

Within an absolute interval

The closest preceding instance within an absolute time interval you chose, or, if none is found, the closest following instance within the absolute time interval you chose.

Regardless of the used matching criteria, if multiple instances of potential predecessor job streams exist in the specified time

interval, the rule used by the product to identify the correct predecessor instance is the following:

1. Tivoli Workload Scheduler searches for the closest instance that precedes the depending job or job stream start time. If such an instance exists, this is the predecessor instance.
2. If there is no preceding instance, Tivoli Workload Scheduler considers the correct predecessor instance as the closest instance that starts after the depending job or job stream start time.

Internetwork dependencies

These are dependencies on jobs or job streams running in another Tivoli Workload Scheduler network. Internetwork dependencies require a network agent workstation to communicate with the external Tivoli Workload Scheduler network.

z/OS

Dependencies in a z/OS environment:

You can have dependencies only between jobs. They can be:

Dependencies between jobs belonging to the same job stream

When you create a job stream, you link, in a sequence, the jobs that it contains. You can also set dependencies between the jobs to have a successor job not begin until its predecessor job is complete. Dependencies determine when the successor job runs.

Dependencies between jobs belonging to different job streams

These dependencies are named *external*. External jobs represent jobs that are part of other job streams. You can create a dependency between jobs in different job streams by creating a dependency on an external job.

“Condition dependencies” on page 51

It is a relationship between one job, named a *conditional successor*, and one or more jobs, named *conditional predecessors*, stating that the conditional successor can run only when a specific combination of conditional predecessor status and return code values occurs. You can define a conditional dependency where the conditional successor starts if its conditional predecessors are in ended-in-error or started status.

Note: Condition dependencies are always managed as external dependencies, even if they link jobs belonging to the same job stream occurrence in the plan.

Job streams in a z/OS environment do not support dependencies on files or prompts.

“Cross dependencies” on page 55

In multiple heterogeneous scheduling environments you can define dependencies on batch activities managed by other Tivoli Workload Scheduler environments. To define a cross dependency on a job running on a different Tivoli Workload Scheduler engine, you must define a dependency on a locally-defined *shadow job* pointing to the remote job instance and running on a *remote engine workstation*. The remote engine workstation manages the communication with the remote engine using an HTTP or HTTPS connection.

Dependencies on resources are supported by Tivoli Workload Scheduler in both the distributed and z/OS environments.

Related tasks:

“Adding a dependency” on page 98

“Removing a dependency” on page 99

Prompt

Distributed

A *prompt* identifies a text message that is displayed to the operator and halts processing of the job or job stream until an affirmative answer is received (either manually from the operator or automatically by an event rule action). After the prompt is replied to, processing continues. You can use prompts as dependencies in jobs and job streams. You can also use prompts to alert an operator that a specific task was performed. In this case, an operator response is not required.

There are the following types of prompt:

global or named

A prompt that is defined in the database as a scheduling object. It is identified by a unique name and can be used by any job or job stream.

local or ad-hoc

A prompt that is defined within a job or job stream definition. It does not have a name, and it is not defined as a scheduling object in the database, therefore it cannot be used by other jobs or job streams.

recovery or abend

A special type of prompt that you define to be used when a job ends abnormally. The response to this prompt determines the outcome of the job or job stream to which the job belongs. A recovery prompt can also be associated to an action and to a special type of job called a *recovery job*.

Related tasks:

“Creating a task to Monitor Prompts” on page 143

“Adding a dependency” on page 98

“Removing a dependency” on page 99

“Designing your Workload” on page 79

“Listing workload objects” on page 114

Resource

A *resource* is either a physical or logical system resource that you use as a dependency for jobs and job streams. A job or job stream with a resource dependency cannot start to run until the required quantity of the defined resource is available.

Related tasks:

“Creating a task to Monitor Resources” on page 144

“Adding a dependency” on page 98

“Removing a dependency” on page 99

“Designing your Workload” on page 79

“Listing workload objects” on page 114

File

Distributed

A *file* is used as a dependency for jobs and job streams. A job or job stream with a file dependency cannot start to run until the file exists with the characteristics defined in the dependency.

Related tasks:

“Creating a task to Monitor Files” on page 141

“Adding a dependency” on page 98

“Removing a dependency” on page 99

“Designing your Workload” on page 79

“Listing workload objects” on page 114

Condition dependencies

z/OS

In Tivoli Workload Scheduler for z/OS, you can specify that jobs are dependent on other jobs. For example, if job A1 must complete before job A2 can start, then A1 is a *predecessor* of A2, and A2 is a *successor* of A1. These relationships between jobs are called dependencies.

When specifying dependencies, you can also define work flows with alternative branches based on conditions, specifically to achieve the same results as using IF/THEN/ELSE statements in the job JCL. You can use both job return code and job status as *conditional logic* elements to determine the start of a successor job. The following example shows how this works.

A condition dependency relationship is set up by using a *condition*.

You can define condition dependencies at the following levels:

Job level

By conditioning the start of the successor to the check on job return code or status of the predecessor.

Step level

By conditioning the start of the successor to a specific step return code of the predecessor.

How condition dependencies work

A *condition dependency* is a specific check of the status or return code of a predecessor job or of the return code of a job step.

The job processing flow is affected by the conditions set and their final status.

The status of a condition is set based on the rule defined and on the statuses of its condition dependencies.

The condition dependency is evaluated only when a path in the plan exists, otherwise the condition dependency remains Undefined until a manual intervention or a rerun is done.

A possible path for the conditional predecessor exists when at least one of the following conditions occurs:

- The job has status **Completed** and a normal successor exists.

- There is at least one conditional successor that has *all* the subsets of conditions, referencing that conditional predecessor, set to **true**, according to the condition rules.

For example:

- A conditional predecessor (Job A) has several conditional successors (Jobs B, C, D)
- Each conditional successor has a set of condition dependencies, relating to job A, that must be satisfied to make it possible for the successor to start.
- Job A runs and changes its status.
- If at least one subset of conditions between job A and one of its successors is **true**, the path in plan exists and all the successors' condition dependencies related to job A are evaluated. Otherwise all condition dependencies are left undefined.

When specifying predecessors in the database, you can define a list of *conditions* by combining single *condition dependencies* on predecessor job status or return code. You cannot define a job both as a conditional and as a normal predecessor of another job. For each condition you can specify one of the following rules:

- At least *n* number of conditions out of all the condition dependencies must be satisfied. This rule corresponds to the OR operator in Boolean logic.
- All the condition dependencies in the list must be satisfied. This rule corresponds to the AND operator in Boolean logic.

At run time, the scheduler evaluates the condition status resulting from the condition dependencies statuses, based on the selected rule. The condition status can be:

True When ALL condition dependencies are true.
If the rule is set to AND
When ALL condition dependencies are true.
If the rule is set to OR (at least *n* condition dependencies must be true)
When at least *n* condition dependencies are true.

False The condition was not satisfied.
If the rule is set to AND
When at least one condition dependency is false.
If the rule is set to OR (at least *n* condition dependencies must be true)
When at least *n* condition dependencies cannot be true.

Undefined
When the rule cannot be evaluated yet.

A set of conditions results as satisfied if all the conditions are satisfied, according to the logic of the AND operator.

When a predecessor ends, the successor job status changes to one of the following statuses:

Waiting
Undefined, until the scheduler evaluates all the defined conditions. At least one normal predecessor is not in **Completed** or **Suppressed by Condition** status or at least one condition is U (Undefined). The scheduler processes all subsequent statuses as usual, up to a final status.

Ready Ready, when all the defined conditions are satisfied. Job normal predecessors are in **Completed** or **Suppressed by Condition** status and all its conditions are **True**. The scheduler processes all subsequent statuses as usual, up to a final status.

Suppressed by Condition

Suppressed by condition, when the defined condition dependency is not satisfied. At least one condition is **False**.

Note: When evaluating conditional successors status, predecessor jobs in status **Suppressed by Condition** are considered equal to predecessor operations in status **Completed**.

Examples of condition dependencies

Use a job-level condition dependency when you want a successor job to start depending on a combination of one or more return codes or statuses of predecessor jobs.

Figure 3 shows the two different types of job level conditions, one based on the predecessor return code and the other based on the predecessor job status. For example, using the return code as condition type, you can define that job OP2 is dependent on job OP1, specifying that OP2 must run when OP1 ends with a return code in the range from 1 to 3. Similarly, using job status as condition, you can define job OP4 as dependent on job OP3, specifying that OP4 must run if OP3 ends with status **Error**.

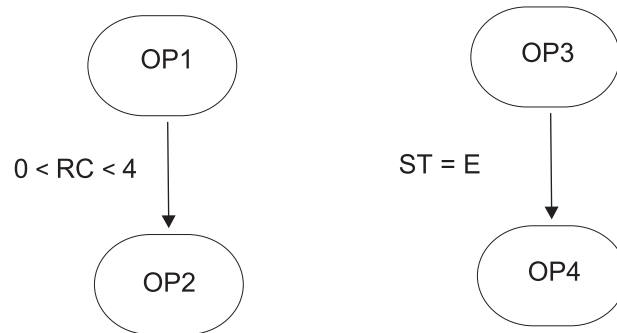


Figure 3. Example of a condition dependency definition

In this example, OP1 is a *conditional predecessor* of OP2 and OP3 is a *conditional predecessor* of OP4.

In the previous example, if OP1 ends with return code of 8, the scheduler sets OP2 to status **Suppressed by Condition**, because the condition is not satisfied.

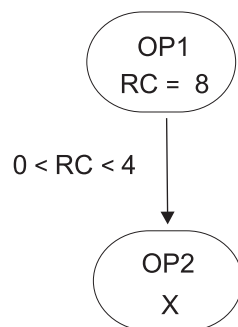


Figure 4. Example of a condition dependency at run time

For further information about conditional logic, see *Tivoli Workload Scheduler for z/OS Managing the Workload*

Step level dependency

If you configured Tivoli Workload Scheduler for z/OS to track step-end events, then the step dependencies are checked at step-end time when the return code value is available.

This section contains an example showing how job processing flow is affected when using step-level conditions.

If the predecessor job is associated to a job comprising several steps, you can specify a dependency on step return codes. Figure 5 shows an example of conditional dependency logic at job step level, to obtain auto-recovery applications with recovery jobs that can start without waiting for the end of predecessor jobs, depending on the result of specific steps.

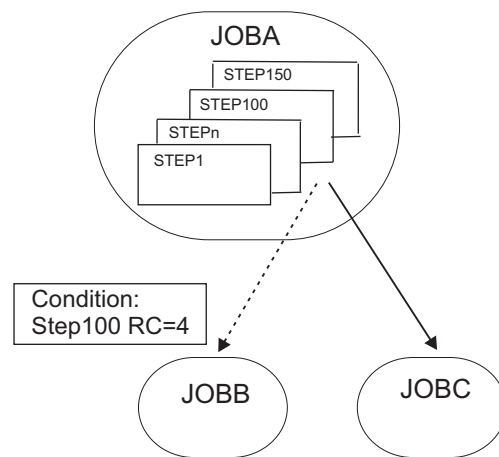


Figure 5. Auto-recovery job stream with step level dependency

In this example:

- JOBB can start if STEP100, belonging to JOBA, ends with RC=4.
- JOBC is a normal successor of JOBA and therefore starts if JOBA status is **Completed**.

Handling recovery using condition dependencies

Using condition dependencies, the error status of a job can be used as a criteria for starting a successor, when this successor is used as a recovery job.

By specifying the conditional recovery job option, you can define that the job is used as the recovery job for a conditional predecessor.

Any conditional predecessor that Ended-in-Error, with a status or error code matching a condition dependency defined for the job, does not prevent the daily plan process from removing the occurrence to which the predecessor belongs. To check if the status **Ended-in-error** can be ignored at occurrence removal phase, the daily plan process uses a field automatically set by the scheduler, corresponding to **Recovered by condition**.

Note: As soon as a recovery job becomes ready, the scheduler checks the predecessors in error status at that time. Any predecessor that ends in error after the recovery job runs cannot be flagged as **Recovered by condition**. The daily plan process removes the occurrence in the following cases:

- The occurrence status is **Completed**.
- The occurrence status is **Ended-in-error**, and includes only jobs in one of the following statuses:
 - **Completed**
 - **Suppressed by condition**
 - **Ended-in-error** with the **Recovered by condition** option specified.

For example, suppose that either JOBR1 or JOBR2 must run when JOBB ends with an error. You can specify JOBB as their conditional predecessor, as shown in Figure 6.

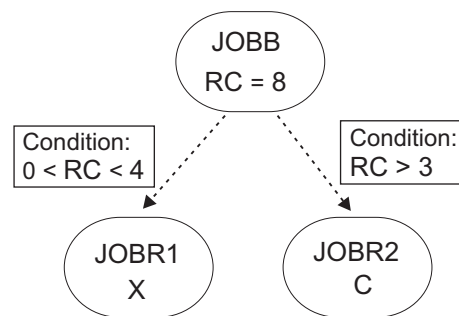


Figure 6. Example of recovery job with condition dependencies

When defining JOBR1 and JOBR2 and specifying JOBB as conditional predecessor, you can also set the **Conditional recovery job** option to have the daily plan process remove the occurrence containing JOBB, because it ended with an error code matching one of the defined condition dependencies.

Cross dependencies

A cross dependency is a dependency of a *local job* on a *remote job* running in a different scheduling environment. It is achieved by using a *shadow job*, which runs in the same environment as the local job and maps the remote job processing.

Cross dependencies help you integrate workload running on more than one engine. They can be both Tivoli Workload Scheduler for z/OS engines (controller) and Tivoli Workload Scheduler engines (master domain manager).

The following objects allow you to define and manage cross dependencies:

Remote engine

A workstation that represents locally a remote Tivoli Workload Scheduler engine. It is a workstation used to run only *shadow jobs*. A shadow job is a job that runs locally and is used to map another job running on a remote engine. This relationship between the two jobs is called a *cross dependency*. You define a remote engine workstation if you want to federate your environment with another Tivoli Workload Scheduler environment, either distributed or z/OS, to add and monitor dependencies on jobs running in the other scheduling environment. This type of workstation uses a connection based on HTTP protocol to allow the two environments to communicate.

Shadow job

A job running locally that is used to map a job running on the remote engine. This job is called a *remote job*. Shadow jobs can run only on remote engine workstations. The shadow job definition contains all the information needed to correctly match the remote job in the plan of the remote engine. The status transition of the shadow job reflects the status transition of the remote job.

Remote job

A job that runs on a remote scheduling environment and is mapped by a shadow job to become a dependency for a job that runs in a local environment.

To add a cross dependency to a local job on a job that is defined on a remote engine, you must define a normal dependency for your local job on a shadow job that:

- Points to the remote job on which you want to create the cross dependency
- Is defined on a local workstation of remote engine type, that points to the engine where the remote job is defined.

To do this, you must

1. Create a remote engine workstation where the shadow job runs.
2. Create a shadow job pointing to a specific job instance defined on a remote engine.

Shadow jobs can be added to the plan by the plan creation process or dynamically at run time. The shadow job scheduled time identifies the remote job instance in the remote engine plan.

The **bind** process is the process to associate a shadow job with a job instance in the remote engine plan.

As soon as the bind is established, the remote engine sends back an HTTP notification containing the status of the bind and, if the bind was successful, the information to identify the remote job instance bound. This information is saved in the shadow job instance details.

3. Add the shadow job as a dependency of the local job.

The resolution of the cross dependency depends on the status of the shadow job, which reflects at any time the status of the remote job. Because the remote job status transition is mapped into the shadow job status transition, the status of the cross dependency is represented by the status of the normal dependency.

The key attributes to identify the remote job instance and the matching criteria depend on the type of remote engine where the remote job instance is defined. z/OS engines support only *closest preceding* matching criteria. Distributed shadow jobs, instead, support the four matching criteria available for external dependencies. See “Dependencies” on page 48 for more details.

The scheduled time of the job stream containing the shadow job is used to find the match.

To avoid incongruence, at plan creation or extension time, consistency checks are performed to ensure that no mismatch has occurred in between the definition of jobs and workstations in the database and their inclusion in the current plan.

Figure 7 summarizes how cross dependencies work.

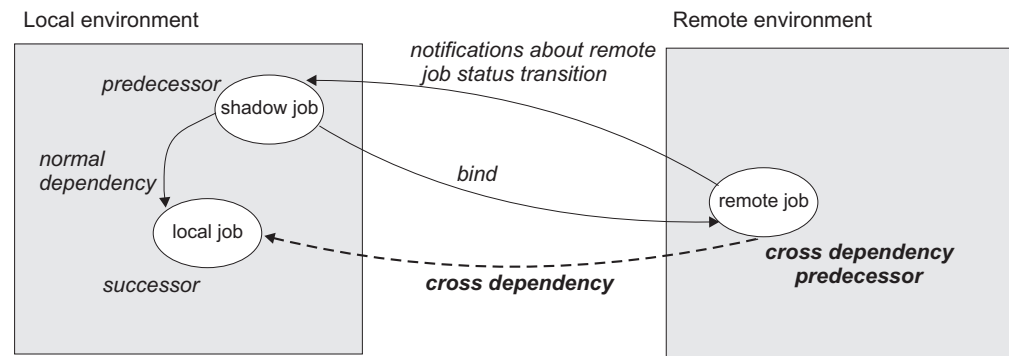


Figure 7. Cross dependencies

For more information about cross dependencies, see the sections about defining and managing cross dependencies in *Tivoli Workload Scheduler User's Guide* and *Reference* and in *Tivoli Workload Scheduler for z/OS Managing the Workload*

Related tasks:

“Creating cross dependencies” on page 99

User

Distributed

A *User* is the user name used as the login value for several operating system job definition. Users must be defined in the database.

If you schedule a job on an agent, on a pool, or on a dynamic pool, the job runs with the user defined on the pool or dynamic pool. However, the User must exist on all workstations in the pool or dynamic pool where you plan to run the job.

Related tasks:

“Designing your Workload” on page 79

“Listing workload objects” on page 114

Workstation class

A *workstation class* is a group of workstations with similar job scheduling characteristics. Any number of workstations can be grouped in a class, and a workstation can be included in many classes. Jobs and job streams can be assigned to run on a specific workstation class and this makes the running of jobs and job streams across multiple workstations easier.

For example, you can set up the following types of workstation classes:

- Workstation classes that group workstations according to your internal departmental structure, so that you can define a job to run on all the workstations in a department
- Workstation classes that group workstations according to the software installed on them, so that you can define a job to run on all the workstations that have a particular application installed
- Workstation classes that group workstations according to the role of the user, so that you can define a job to run on all the workstations belonging to, for example, managers

In the previous example, an individual workstation can be in one workstation class for its department, another for its user, and several for the software installed on it.

Distributed Workstations can also be grouped into domains when your network is set up. Because the domain name is not one of the selection criteria used when choosing where to run a job, you might need to mirror your domain structure with workstation classes if you want to schedule a job to run on all workstations in a domain.

Related tasks:

“Designing your Workload” on page 79

“Listing workload objects” on page 114

Variable table

A *variable table* is a table containing multiple variables and their values. All global parameters, now called *variables*, are contained in at least one variable table.

You are not required to create variable tables to be able to use variables, because the scheduler provides a default variable table.

However, you might want to define a variable with the same name, but different values, depending on when and where it is used. You do this by assigning different values to the same variable in different variable tables. You can then use the same variable name in different job definitions and when defining prompts and file dependencies. Variable tables can be assigned at run cycle, job stream, and workstation level.

You can also define verification criteria and dependency list and associate them to variables.

Variable tables can be particularly useful in job definitions when a job definition is used as a template for a job that belongs to more than one job stream. For example, you can assign different values to the same variable and reuse the same job definition in different job streams.

Related tasks:

“Designing your Workload” on page 79

“Listing workload objects” on page 114

Workload Broker job definition

A *Workload Broker job definition* is a text file in Job Submission Description Language (JSDL) schema file format, which contains all the parameters necessary to run a job.

The Job Brokering Definition Console provides an easy-to-use graphical interface that you can install locally to create and edit job definitions. Job definitions, saved in the JSDL schema, are created by your input into the Job Brokering Definition Console.

The JSDL schema offers great flexibility and supports a wide variety of criteria to determine job requirements, scheduling and load-balancing.

Examples of how to define a job to achieve different objectives are provided in *Tivoli Workload Scheduler Scheduling Workload Dynamically*.

From the Dynamic Workload Console, you can create Workload Broker job definitions that map Workload Broker jobs. You can also monitor Workload Broker jobs and browse the corresponding job logs.

Related tasks:

“Creating Workload Broker objects” on page 107

Production process

Tivoli Workload Scheduler production is based on a plan that runs in a production period.

You can define the production period when creating or extending the *production plan* and it can span from a few hours to multiple days (by default it lasts 24 hours).

The production plan contains information about the jobs to run, on which fault-tolerant agent, and what dependencies must be satisfied before each job can start.

Distributed You use the JnextPlan script to generate the production plan and distribute it across the Tivoli Workload Scheduler network. Then, if you want to extend your production plan at a fixed time interval, for example every day, you have the option to automate the extension by using the final job stream at the end of each production period. A sample job stream helps you to automate plan management and runs the sequence of script files included in JnextPlan to generate the new production plan.

When the production plan is generated, all of the required information about that production period is taken from the scheduling environment and object definitions and is included in the plan.

During the production period, the production plan is regularly updated to show what work is completed, in progress, and left to process.

In Tivoli Workload Scheduler for distributed environments or in a z/OS end-to-end network, a file called *Symphony* contains all the information about the production plan. This file is sent to all the subordinate domain managers and fault-tolerant agents in the scheduling environment. This allows the fault-tolerant agents throughout the network to continue their processing even if the network connection to their domain manager is down.

Tivoli Workload Scheduler processes monitor the production plan and make calls to the operating system to launch jobs as required. The operating system runs the jobs, and informs Tivoli Workload Scheduler if the job completed successfully. This information is used to update the production plan to indicate the status of the job.

From the Dynamic Workload Console or the command line interface, you can view and make changes in the current production plan.

Database

The Tivoli Workload Scheduler database, hereafter referred to as the database, is a relational database that is accessible by the master domain manager and contains all the definitions for scheduling objects, such as jobs, job streams, resources, and

workstations. It also holds statistics of job and job stream execution, as well as information about the user ID that created an object and when an object was last modified.


For more information about the types and versions of the supported relational database, see the Tivoli Workload Scheduler documentation.

Related tasks:

“Designing your Workload” on page 79

Plans

A plan contains all jobs and job-related scheduling objects that are scheduled for a selected time interval. There are different types of plans based on the type of Tivoli Workload Scheduler environment you are connected to.

Note:  The only type of plan that you can access through the Dynamic Workload Console is the current plan.

The following plans are available:

Production plan (current plan)

A production plan (in distributed environment) or current plan (in z/OS environment) is the master control for all job scheduling activity planned for a user-defined time interval, named the *production period*. Scheduling object definitions stored in the database, such as jobs and job streams, become instances in the production plan, where they can be monitored and modified.

The production plan is created on the master domain manager and contains all the jobs and job streams that are scheduled to run within the production period together with their depending objects and all workstation definitions. The production plan can be extended to cover future time intervals. Any job streams that did not complete successfully within the production period or that are either running or still waiting to be run, can be carried forward into the plan extension.

The production plan data is stored in the Symphony file and replicated in the database. With Tivoli Workload Scheduler version 9.1, accessing this information from the Dynamic Workload Console directly queries the database thereby improving response times.

Preproduction plan

A preproduction plan is used to identify in advance the job stream instances and the job stream dependencies involved in a specified time period.

This improves performance when generating the production plan by preparing in advance a high-level schedule of the anticipated production workload.

The preproduction plan contains:

- The job stream instances to be run during the covered time period.
- The external dependencies that exist between the job streams and jobs included in different job streams.

Symnew plan

A Symnew plan is a temporary plan. It is an intermediate production plan that covers the whole time the new production plan that is being generated will cover. It is replaced by the production plan as soon as it starts.

Archived Plan

An archived plan is a copy of an old production plan that ran in the Tivoli Workload Scheduler environment and that is now stored in the Tivoli Workload Scheduler database.

Using this type of plan you can, for example, see the result of running a past production plan. The difference between using an archived plan and a forecast plan covering the same time interval is that an archived plan shows how the real production was based on the job and job stream processing results, while a forecast plan shows how the production was planned to be.

Trial plan

A trial plan is a projection of what a production plan would be if it covered a longer period of time. For example, if you generate a production plan that covers two days, but you want to know what the plan would be if it covered three days, you can generate a trial plan.

A trial plan is typically created to extend a production plan and to have an idea of future impacts on the scheduling environment. Therefore, if there is a valid production plan, the start time option is greyed out. By default, the trial plan start date is the same as the production plan end date.

Using this type of plan you can, for example, see how the current production evolves based on the job and job stream dependencies defined in the production plan, if available, or in the preproduction plan. Trial plans are based on the information contained either in the production or in the preproduction plan. If neither is available, a trial plan cannot be created.

Forecast plan

A forecast plan is a projection of what the production plan would be in a chosen time interval. For example, if you generate a production plan that covers two days and you want to know what the plan would be for the next week you can generate a forecast plan.

A forecast plan is typically created to anticipate and solve any kind of scheduling problems, therefore the start time is always enabled and it is a mandatory field.

Using this type of plan you can, for example, see how the production will be in a future time interval based on the job and job stream dependencies defined in the Tivoli Workload Scheduler database. Based on this information, you can modify some information in the database, if needed, before extending the production plan.

When workload service assurance is enabled, it can calculate the predicted start time of each job in the job stream. You can enable and disable this feature using the **enForecastStartTime** global option. Tivoli Workload Scheduler calculates the average run for each job based on all previous runs. For complex plans, enabling this feature could negatively impact the time taken to generate the forecast plan.

Note: Neither the trial nor the forecast plan takes into account any dynamic updates made to the Symphony file while the production plan is being processed. Therefore, all the job streams it contains are in one of the following states:

HOLD

If they are dependent on other job streams or if their start time is later than the plan start time.

READY

If they are free from dependencies and their start time has elapsed.

Related concepts:

“Selecting the working plan” on page 155

“Monitoring the progress of your plan” on page 123

Request a graphical view of the progress of the current plan.

Related tasks:

“Creating a task to Show Plan View” on page 157

“Generating Trial and Forecast Plans” on page 157

Related information:

“Graphical Views in the plan” on page 210

Preproduction plan

The preproduction plan is used to identify in advance the job stream instances and the job stream dependencies involved in a specified time period.

This improves performance when generating the production plan by preparing in advance a high-level schedule of the predicted production workload.

The preproduction plan contains:

- The job stream instances to be run during the time interval of the plan.
- The external follows dependencies that exist between the job streams and jobs included in different job streams.

A job or job stream that cannot start before another specific external job or job stream is successfully completed is named *successor*. An external job or job stream that must complete successfully before the successor job or job stream can start is named *predecessor*.

Tivoli Workload Scheduler automatically generates, expands, and updates, if necessary, the preproduction plan by completing the following steps:

- Removes any job stream instances that are in COMPLETE or CANCEL state.
- Selects all the job streams that are scheduled to run after the end of the current production plan and generates their instances.
- Resolves all job and job stream dependencies, including external follows dependencies, according to the defined matching criteria.

To avoid any conflicts, the database is locked during the generation of the preproduction plan and unlocked when the generation completes or if an error condition occurs.

At this stage only the job streams with the time they are scheduled to start and their dependencies are highlighted. All the remaining information about the job streams and the other scheduling objects (calendars, prompts, domains, workstations, resources, files, and users) that will be involved in the production plan for that time period are not included, but are retrieved from the database as soon as the production plan is generated.

When the production plan is extended, old job stream instances are automatically removed. The criteria used to remove these instances is based on the following conditions:

- The first job stream instance that is not in COMPLETE state when the new plan is generated (FNCJSI). This job stream instance can be both a planned instance,

that is an instance added to the plan when the production plan is generated, and a job stream instance submitted from the command line during production using the **conman sbs** command.

- The time period (T) between the time FNCJSI is planned to start and the end time of the old production plan.

Assuming T is this time period, the algorithm used to calculate which job stream instances are removed from the preproduction plan is the following:

if T < 7

All job stream instances older than 7 days from the start time of the new production plan are removed from the preproduction plan; all job stream instances closer than 7 days to the start time of the new production plan are kept regardless of their states.

if T > 7

All job stream instances older than FNCJSI are removed from the preproduction plan; all job stream instances younger than FNCJSI are kept.

This algorithm is used to ensure that the preproduction plan size does not increase continuously and, at the same time, to ensure that no job stream instance that is a potential predecessor of a job stream newly added to the new preproduction plan is deleted.

For more information about how you can open the preproduction plan in view mode from the Dynamic Workload Console, see the section about how to view preproduction plan in the Dynamic Workload Console User's Guide.

Note: In the Tivoli Workload Scheduler for z/OS® terminology the concept that corresponds to the preproduction plan is *long term plan* (LTP).

Related tasks:

“View preproduction plan” on page 159

Engine connections

An engine connection is a set of configuration information that identifies a specific workstation in the Tivoli Workload Scheduler environment in the network.

To manage scheduling objects you must connect from the Dynamic Workload Console to a Tivoli Workload Scheduler environment. In Dynamic Workload Console you do this by defining engine connections.

You can connect to both Tivoli Workload Scheduler distributed and z/OS environments and you can create as many engine connections as you need.

When you create an engine connection, you give it a name and select the plan that you want it to access. The selected plan must be accessible from that workstation.

z/OS If you connect to a Tivoli Workload Scheduler for z/OS environment, the plan that you access is the current plan and the engine that you connect to is the controller workstation, which is the management hub of the Tivoli Workload Scheduler for z/OS environment.

Distributed If you connect to a Tivoli Workload Scheduler distributed environment, you can access different types of plans and connect to different types of engines. Based on the type of plan that you select and the engine that you connect to, you

can get different results when running your tasks in the same Tivoli Workload Scheduler distributed environment. You can connect to:

The master domain manager workstation

The top management hub. Select this workstation if you want to access the entire set of objects involved in the current plan, or if you want to access a trial plan, a forecast plan, or an archived plan. You can define and use different engine connections to the master domain manager, each accessing a different plan.

A fault-tolerant agent if the connector is installed

A workstation where jobs and job streams are run. Select this workstation if you want to access the set of objects involved in the current plan and scheduled to run in that fault-tolerant agent. You choose this option if you need up-to-date information regarding job status on this workstation.

Related tasks:

“Creating and managing engine connections” on page 8

Event management

You can use the *event management* feature to launch a predefined set of actions in response to events that occur on the nodes where Tivoli Workload Scheduler runs.

The main elements of event management are:

- “Events”
- “Actions” on page 65
- “Event rules” on page 66

You can use event management capabilities to:

- Create Event Rules
- Create and run Workload Events tasks

Events

An event represents a set of circumstances that match selected criteria. Events are divided into the following major categories:

Tivoli Workload Scheduler objects related events

All the events relating to scheduling objects such as jobs, job streams, workstations, and prompts. This type of event is described in more detail in Tivoli Workload Scheduler plan events.

Note: Any change performed on a workstation referenced in a rule is not reported in the rule. For example if you modify, update, or delete a workstation that is referenced in a rule, the rule ignores the change and continues to consider the workstation as it was when it was included in the rule.

File monitoring events

Events relating to changes to files and logs. File monitoring events are not supported on IBM i systems.

This type of event is described in more detail in File monitor.

Application monitoring events

Events relating to Tivoli Workload Scheduler processes, file system, and message box. Application monitoring events are not supported on IBM i systems.

This type of event is described in more detail in Application Monitor.

SAP related events

These events are available only if you have installed Tivoli Workload Scheduler for Applications and they are generated by external SAP systems. This type of event is described in more detail in SAP Monitor.

Generic events

Events used to manage custom events sent by external applications. You can write an XML file to define a custom event. A schema is provided to validate your XML, as well as a basic event template that you can use as a starting point. For more information, see the schemas for generic events. Events of this category are:

- Changes in a resource of the operating system, such as processes and memory
- Email received

Actions

When one or more of the above events occurs, you can specify which actions to perform. Actions are divided into the following main categories:

Operational actions

Actions that cause a change in the status of one or more Tivoli Workload Scheduler objects. Actions in this category include:

- Submitting jobs or job streams
- Submitting ad hoc jobs
- Replying to a prompt

This type of action is described in more detail in Tivoli Workload Scheduler actions.

- Adding an application occurrence (job stream) to the current plan on Tivoli Workload Scheduler for z/OS in Tivoli Workload Scheduler-Tivoli Workload Scheduler for z/OS end-to-end scheduling configurations.

This type of action is described in more detail in Tivoli Workload Scheduler for z/OS actions.

Notification actions

Actions such as:

- Sending emails or SMS. For details, see Mail sender plug-in
- Forwarding Tivoli Enterprise Console events. For details, see Message logger
- Writing messages in a logging repository. For details, see Tivoli Enterprise Console event forwarder
- Forwarding events to a Tivoli Business Services Manager server. For details, see Tivoli Business Services Manager Console event forwarder.
- Opening a ticket to a SmartCloud Control Desk. For details, see SmartCloud Control Desk.

Generic actions

Actions performed by running a command. This type of action is described in more detail in Generic action plug-in.

Event rules

Use *event rules* to associate one or more events to the response actions that you want to perform. When you create an event rule, you are actually creating an *event rule definition* in the database. While the event rule is in **Draft** status, it is not deployed to the Tivoli Workload Scheduler. All new and modified non-draft rules saved in the database are periodically (by default every five minutes) found, built, and deployed by an internal process named rule builder. At this time they become active. Meanwhile, an event processing server, which is normally located in the master domain manager, receives all events from the agents and processes them. The updated monitoring configurations are downloaded to the Tivoli Workload Scheduler agents and activated. The occurrence of an event rule that has performed the corresponding actions is called the *event rule instance*.

Related concepts:

“Event management configuration” on page 9

Reports

Create a *report* task to customize and generate Tivoli Workload Scheduler reports, which you can then view, print, and save, in different kinds of output. Reports help you in many business-related activities, such as:

Tuning the workload on the workstations

- Workstation Workload Summary
- Workstation Workload Runtimes

Extract detailed information about the plan

- Planned Production Details
- Actual Production Details

Detect jobs with exceptions

- Job Run History
- Job Run Statistics

The following table shows the available reports and their details.

Table 7. Report types

Report Name	Description	Output
Job Run History	Collects the historical job run data during a specified time interval. Use it to detect which jobs ended in error or were late. It also shows which jobs missed their deadline, long duration jobs, and rerun indicators for reruns.	Available in table format only. It can be HTML or CSV.
Job Run Statistics	Collects the job run statistics. Use it to detect success, error rates; minimum, maximum, and average duration; late and long duration statistics.	Available in graphic and table formats. It can be HTML or CSV.
Workstation Workload Summary	Shows the workload on the specified workstations. The workload is expressed in terms of number of jobs that ran on them. It helps for capacity planning adjustments (workload modelling and workstation tuning).	Available in graphic and table formats. It can be HTML or CSV.

Table 7. Report types (continued)

Report Name	Description	Output
Workstation Workload Runtimes	Shows job run times and duration on the specified workstations. It helps for capacity planning adjustments (workload modelling and workstation tuning).	Available in graphic and table formats. It can be HTML or CSV.
Custom SQL	Allows you to create reports that best fit your business needs. You can specify an SQL query or import SQL scripts.	Available in table format only. It can be HTML or CSV.
Planned Production Details	Extracts information about planned production plans into either an XML or a CSV format, to be used respectively, with Microsoft Project and Microsoft Excel. This also allows users who do not know Tivoli Workload Scheduler to access plan information in a familiar format.	Available in CSV or XML format only.
Actual Production Details	Extracts current plan information into either an XML or a CSV format, to be used, respectively, with Microsoft Project and Microsoft Excel. This also allows users who do not know Tivoli Workload Scheduler to access plan information in a familiar format.	Available in CSV or Microsoft Project format only.

The output of historical reports, which is extracted from the database, consists of the following main sections. The output of a Planned and Actual report is not structured because it is a file that must be opened with an external program.

Report header

Contains the report title, description, engine name, engine type, creation time, type, and the total number of the result set extracted.

Report of content

Contains a set of hyper-links to each section and subsection.

Report format

Depending on the kind of information you are processing, you can choose to view it in the most appropriate format. The report output can be in:

Table format

It shows information organized in rows and columns, in a CSV or HTML file.

Graphical format (HTML)

If you choose the graphical formats, depending on the report type, and on the information you choose to include, you can have data displayed in pie charts, bar charts, line charts, or tables.

Note: To see report output correctly, make sure that you configure your browser as follows:

- Allow pop-up windows.
- Remove any optional browser toolbar that you installed, if its settings prevent new windows from opening.
- To see CSV reports, configure the browser security settings to automatically prompt for file downloads.

Related reference:

Chapter 14, "Reporting," on page 165

Workload service assurance

Workload service assurance is an optional feature that allows you to identify critical jobs and to ensure that they are processed in a timely manner.

When the workload service assurance feature is enabled, you can indicate that a job is critical and define a deadline by which it must be completed when you add the job to a job stream. Defining a critical job and deadline triggers the calculation of timings for all jobs that make up the critical network. The critical network includes the critical job itself and any predecessors that are defined for the critical job. When changes that have an impact on timings are made to the critical network, for example addition or removal of jobs or follows dependencies, the critical start times are automatically recalculated.

The critical network is constantly monitored to ensure that the critical job deadline can be met. When a critical network job completes, timings of jobs that follow it are recalculated to take account of the actual duration of the job. The system also acts automatically to remedy delays by prioritizing jobs that are actually or potentially putting the target deadline at risk. Some conditions that cause delays might require your intervention. A series of specialized critical job views, available on the Dynamic Workload Console, allow you to monitor critical jobs, display their predecessors and the critical paths associated with them, identify jobs that are causing problems, and drill down to identify and remedy problems.

Dynamic critical path

If a job is critical and must complete by the deadline set on the database you can mark it as a *critical job* thus specifying that it must be considered as the target of a *critical path*. The critical path consists of the critical job predecessors with the least *slack time*. In a critical job predecessor path, the slack time is the amount of time the predecessor processing can be delayed without exceeding the critical job deadline. It is the spare time calculated using the deadline, scheduled start, and duration settings of predecessors jobs. The calculation of critical path is performed dynamically. In this way, during daily planning processing, a critical path including the internal and external predecessors of the critical job is calculated, and a table of predecessors is cached (in the local memory for z/OS and on the master domain manager for distributed systems). Every time a predecessor of the critical job starts delaying, the scheduler *dynamically* recalculates the critical path, to check whether a new path, involving different jobs, became more critical than the path calculated at daily planning phase.

You can launch a query for all the jobs included in a critical path by clicking **Critical Path** in the panels that show the results of monitor jobs tasks.

As well as jobs included in the critical path job list, there are other lists of jobs that might be important to manage to ensure that your critical job does not fail.

Hot list

The *hot list* contains a subset of critical predecessors that can cause a delay to the critical job because they are states such as error, late, fence (for distributed systems only), suppressed (for distributed systems only), or long duration. If these jobs do not complete successfully on time, they prevent the critical job from completing on

time. Using the hot list view, you can quickly see which jobs need you to take appropriate recovery actions. Jobs included in the hot list are not necessarily also included in the critical path.

You can launch a query for all the jobs in the hot list by clicking **Hot List** in the panels that show the results of monitor critical jobs tasks.

Related tasks:

“Creating a task to Monitor Critical Jobs” on page 134

Related reference:

“Limit the number of objects retrieved by queries” on page 190

Related information:

“Using workload service assurance to monitor z/OS critical jobs” on page 173

Processing and monitoring critical jobs

Automatic tracking and prioritizing of critical network jobs.

Workload service assurance provides automatic tracking and prioritizing of critical network jobs and online functions that allow you to monitor and intervene in the processing of critical network jobs.

Automatic tracking and prioritizing

To ensure that critical deadlines can be met, workload service assurance provides the following automated services for critical jobs and for predecessor jobs that form their critical networks:

Promotion

When the critical start time of a job is approaching and the job has not started, the promotion mechanism is used. A promoted job is assigned additional operating system resources and its submission is prioritized.

The timing of promotions is controlled by the global option `promotionoffset`. Promoted jobs are selected for submission after jobs that have priorities of "high" and "go", but before all other jobs. Prioritizing of operating system resources is controlled by the local options `jm promoted nice` (UNIX and Linux) and `jm promoted priority` (Windows).

Calculation of the critical path

The critical path is the chain of dependencies, leading to the critical job, that is most at risk of causing the deadline to be missed at any given time. The critical path is calculated using the estimated end times of the critical job predecessors. Working back from the critical job, the path is constructed by selecting the predecessor with the latest estimated end time. If the actual end time differs substantially from the estimated end time, the critical path is automatically recalculated.

Figure 8 on page 70 shows the critical path through a critical network at a specific time during the processing of the plan.

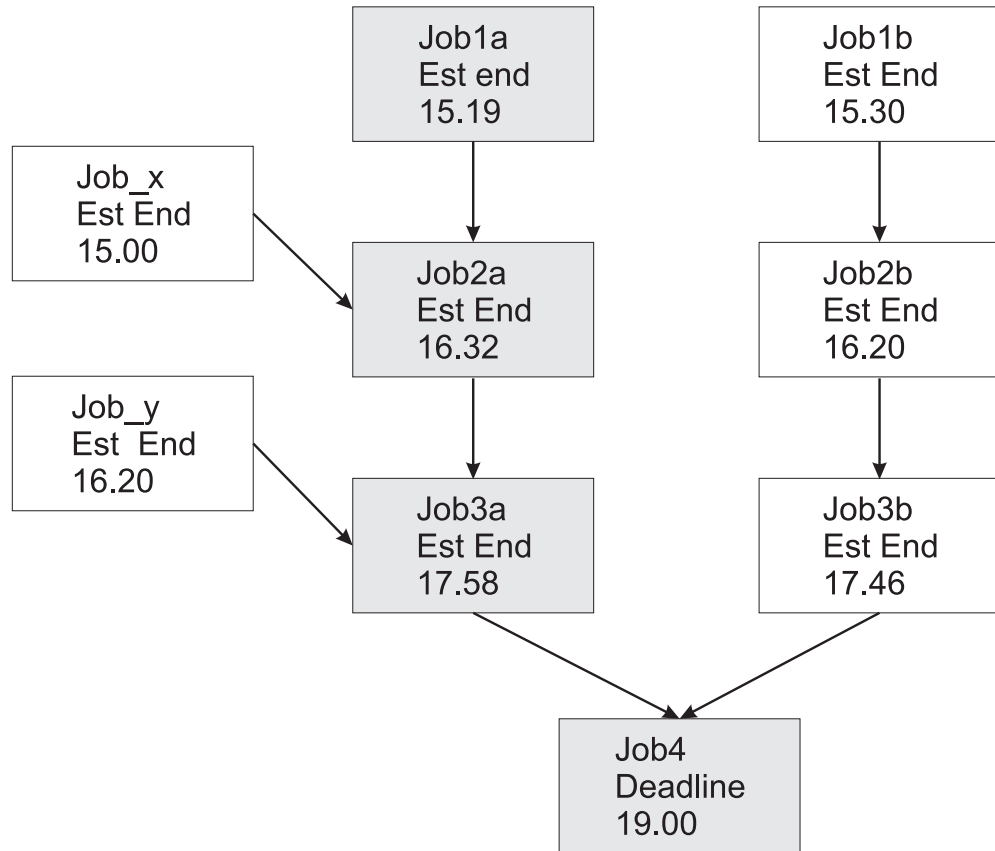


Figure 8. Critical path

At this specific time, the critical path includes Job3a, Job2a, and Job1a. Job3a and Job3b are the immediate predecessors of the critical job, job4, and Job3a has the later estimated end date. Job3a has two immediate predecessors, Job2a and Job_y. Job2a has the later estimated end time, and so on.

Addition of jobs to the hot list

Jobs that are part of the critical network are added to a hot list that is associated to the critical job itself. The hot list includes any critical network jobs that have a real or potential impact on the timely completion of the critical job. Jobs are added to the hot list for the one or more of the following reasons. Note that only the jobs that begin the current critical network, for which there is no predecessor, can be included in the hot list.

- The job has stopped with an error. The length of time before the critical start time is determined by the `approachingLateOffset` global option.
- The job has been running longer than estimated by a factor defined in the `longDurationThreshold` global option.
- The job has still not started, although all its follows dependencies have either been resolved or released, and at least one of the following conditions is true:
 - The critical start time has nearly been reached.
 - The job is scheduled to run on a workstation where the limit is set to zero.
 - The job belongs to a job stream for which the limit is set to zero.
 - The job or its job stream has been suppressed.

- The job or its job stream currently has a priority that is lower than the fence or is set to zero.

Setting a high or potential risk status for the critical job

A critical job can be set to the following risk statuses:

High risk

Calculated timings show that the critical job will finish after its deadline.

Initially, estimated start and end times are used. As jobs are completed, timings are recalculated to take account of the actual start and end times of jobs.

Potential risk

Critical predecessor jobs have been added to the hot list.

Online tracking of critical jobs

The Dynamic Workload Console provides specialized views for tracking the progress of critical jobs and their predecessors. You can access the views from the Dashboard or by creating Monitor Critical Jobs tasks.

The initial view lists all critical jobs for the engine, showing their status: normal, potential risk, or high risk. From this view, you can navigate to see:

- The hot list of jobs that put the critical deadline at risk.
- The critical path.
- Details of all critical predecessors.
- Details of completed critical predecessors.
- Job logs of jobs that have already run.

Using the views, you can monitor the progress of the critical network, find out about current and potential problems, release dependencies, and rerun jobs.

Planning critical jobs

Planning critical jobs.

Workload service assurance provides the means to identify critical jobs, define deadlines, and calculate timings for all jobs that must precede the critical job.

If it is critical that a job must be completed before a specific time, you can flag it as critical when you add it to a job stream using the Workload Designer functions on the Dynamic Workload Console. You can define the deadline either at job or job stream level.

Jobs can also be flagged as critical by including the **critical** key word on the job statement when you create or modify a job stream using the **composer** command line.

When you run the command to include the new job in the production plan, all jobs that are direct or indirect predecessors of the critical job are identified. These jobs, together with the critical job itself, form a critical network.

Because timing of jobs in the critical network must be tightly controlled, Time Planner calculates the following timing benchmarks for each critical network job:

Critical start

Applies to distributed systems only and represents the latest time at which the job can start without causing the critical job to miss its deadline.

Critical start times are calculated starting with the deadline set for the critical job and working backwards using the estimated duration of each job to determine its critical start time. For example, if the critical job deadline is 19:00 and the estimated duration of the critical job is 30 minutes, the critical job will not finish by the deadline unless it starts by 18:30. If the immediate predecessor of the critical job has an estimated duration of 20 minutes, it must start at latest by 18:10.

Note: Only the deadline of the critical job is considered when calculating critical start times for jobs in the critical network. If other jobs have deadlines defined, their critical start times might be later than their deadlines.

Earliest start

Represents the earliest time at which a job in the critical network can start, taking into consideration all dependencies and resource requirements.

Estimated start times are calculated starting with the earliest time at which the first job or jobs in the critical network can start and working forward using the estimated duration of each job to estimate the start time of the job that follows it.

Estimated start and end time

For the initial calculations, these values are set to the planned start and end time. They are subsequently recalculated to take into consideration any changes or delays in the plan.

Estimated duration

The estimated duration of a job is based on statistics collected from previous runs of the job. If the job has never run before, a default value of one minute is used. Take this into account when considering the accuracy of calculated timings for critical job networks that include jobs running for the first time.

The timings for each job in the critical network are added to the Symphony file, which includes all the plan information and is distributed to all workstations on which jobs are to be run.

As the plan is run, Plan Monitor monitors all critical networks: subsequent changes to the critical network that affect the timing of jobs trigger the recalculation of the critical and estimated start times. Changes might include manual changes, for example, releasing dependencies or rerunning jobs, and changes made automatically by the system in response to a potential or actual risk to the timely completion of the critical job.

Specific views for critical jobs and their predecessors, available from the Dynamic Workload Console, allow you to keep track of the processing of the critical network. The views can immediately identify problems in your planning of the critical job. For example, if the estimated start time of a job in the critical network is later than the critical start time, this is immediately signalled as a potential risk to the critical job.

Tivoli Workload Scheduler for SAP

With SAP R/3 support, you can use Tivoli Workload Scheduler to do the following tasks:

- Use Tivoli Workload Scheduler standard job dependencies and controls on SAP R/3 jobs.
- Create SAP R/3 jobs using the Tivoli Workload Scheduler interface.
- Schedule SAP R/3 jobs to run on specified days and times, and in a defined order.
- Define inter-dependencies between SAP R/3 jobs and jobs that run on different platforms.
- Define the national language support options.
- Use the SAP R/3 Business Warehouse Support function.
- Customize job execution return codes.
- Use SAP R/3 logon groups for load balancing and fault-tolerance.
- Work with SAP R/3 variants and placeholders.
- Use Business Component-eXternal Interface Background Processing (BC-XBP 2.0) interface support to:
 - Intercept jobs
 - Track child jobs
 - Keep all job attributes when you rerun a job
 - Raise events

Note: For more information about SAP, see Tivoli Workload Scheduler for Applications User's Guide.

Scheduling process for the SAP R/3 extended agent

Tivoli Workload Scheduler launches jobs in SAP R/3 by using jobs defined on the following workstations that support the r3batch access method:

- A Tivoli Workload Scheduler extended agent workstation. A workstation that is hosted by a fault-tolerant agent or master workstation.
- A dynamic agent workstation.
- A dynamic pool.
- A z-centric workstation.

These supported workstations use the r3batch access method to communicate with the SAP system. The access method is used to pass SAP R/3 job-specific information to predefined SAP R/3 instances. The access method uses information provided in an options file to connect and launch jobs on an SAP R/3 instance.

Note: For more information about this, see Tivoli Workload Scheduler for Applications User's Guide.

You can define multiple agent workstations to use the same host, by using multiple options entries or multiple options files. Using the SAP R/3 agent name as a key, r3batch uses the corresponding options file to determine which instance of SAP R/3 will run the job. It makes a copy of a template job in SAP R/3 and marks it as able to run with a start time of start immediate. It then monitors the job through to completion, writing job progress and status information to a job standard list found on the host workstation.

Chapter 9. Creating and Editing Objects in the Database

To create and edit objects in the database read the following sections.

Designing your Scheduling Environment

To begin working with Tivoli Workload Scheduler you must design your scheduling environment.

The scheduling environment is composed of the following objects:

- Workstations
- **Distributed** Domains

Depending on whether your engine is distributed or z/OS, the steps to follow are different:

- **Distributed** You create workstations and domains. Depending on the business needs of your organization and the complexity of your network, you can decide to have a hierarchical domain structure.
- **z/OS** You define workstations.

Related concepts:

“Scheduling objects” on page 37

Creating a workstation

To create a workstation definition in the database, perform the steps described in the following sections.

Creating distributed workstations

Distributed

You can create multiple workstation definitions.

For more information about the main workstation types and their attributes, see Workstation.

For more information about the main workstation types and their attributes, see the section about workstation definition in the *Tivoli Workload Scheduler User's Guide and Reference*.

To add a workstation definition to the database and to assign it to a domain, perform the following procedure.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.



1. From the navigation toolbar, click **Administration > Workload Environment Design > Create Workstations**.
2. Select a distributed engine from the list and click **Create Workstations**.

3. In the Workstations properties panel, specify the attributes for the workstation you are creating. Depending on the type of workstation you select, some attributes are mandatory.
4. To assign the workstation to an existing domain or to create a new domain, click **Assign to Domain**.
5. Click **Save**.

The workstation is now added to the database. Alternatively, click **Administration > Workload Environment Design > List Workstations**, select a distributed engine, and click **Display**. From the workstation table, click **New**.

Note: You can add workstation definitions to the database at any time, but you must run **JnextPlan -for 0000** to add the workstation definition to the plan, so that you can run jobs on this-workstation. For dynamic agent workstation definitions, if you have the `enAddWorkstation` global option set to "yes", the workstation definitions are automatically added to the plan after they are defined in the database.

Related concepts:

"Workstation" on page 31

"Domain" on page 34
The domain.

Related reference:

"Workstation types" on page 206

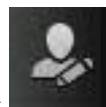
"Type of communication based on SSL communication options" on page 198

Creating z/OS workstations

z/OS

To create a z/OS workstation definition in the database, perform the following steps.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.



1. From the navigation toolbar, click **Administration > Workload Environment Design > Create Workstations**.
2. Select a z/OS engine from the list and click **Create Workstations**.
3. Specify the workstation attributes using the **General**, **Resources**, and **Open Time Intervals** tabs as appropriate. Depending on the type of workstation you select, some attributes are mandatory.
4. Click **Save**.

The workstation is now added to the database. Alternatively, click **Administration > Workload Environment Design > List Workstations** select a z/OS engine, and click **Display**. From the workstation table, click **New**.

Related concepts:

"Workstation" on page 31

Related reference:

"Workstation types" on page 206

Creating z/OS virtual workstations

z/OS

To create a z/OS virtual workstation definition in the database, perform the following steps.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.



1. From the navigation toolbar, click **Administration > Workload Environment Design > Create Workstations**.
2. Select a z/OS engine from the list and click **Create Virtual Workstation**.
3. Specify the workstation attributes using the **General** and **Destinations** tabs as appropriate. Depending on the type of workstation you select, some attributes are mandatory.
4. Click **Save**.

The workstation is now added to the database. Alternatively, click **Administration > Workload Environment Design > List Workstations** select a z/OS engine, and click **Display**. From the workstation table, click **New Virtual**.

Related concepts:

"Workstation" on page 31

Related reference:

"Workstation types" on page 206

Editing workstation definitions

You can manage multiple workstation definitions.

For more information about the main workstation types and their attributes, see the section about workstation definition in the Tivoli Workload Scheduler User's Guide and Reference.

To modify a workstation definition in the database and to assign it to a domain, perform the following steps.



1. From the navigation toolbar, click **Administration > Workload Environment Design > List Workstations** and select an engine.
2. Optionally, specify some filtering criteria that help you find the workstation you want to modify. You can also customize the workstation attribute that you want to display in the columns.
3. Click **Display**. From the workstation table, select a workstation click **Edit** to modify its properties or choose another action from the buttons and menu available.
4. In the Workstations properties panel, specify the attributes for the workstation you are modifying. Depending on the type of workstation you select, some attributes are mandatory.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.

5. To assign the workstation to an existing domain or to create a new domain, click **Assign to Domain**.
6. Click **Save**.

The workstation definition is now modified to the database.

Note: You can add workstation definitions to the database at any time, but you must run JnextPlan -for 0000 again to be able to run jobs on newly-created workstations. Every time you run JnextPlan all workstations are shut down and restarted.

Creating a domain

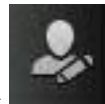
Distributed

You add domain definitions to the Tivoli Workload Scheduler database in either of the following ways:

- When you are creating a distributed workstation, from the **Create Workstations** panel.
- When displaying the list of workstations, by clicking **Domains view**.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.

To create a new domain when creating a workstation, perform the following steps:



1. From the navigation toolbar, click **Administration > Workload Environment Design > Create Workstations**.
2. Select a distributed engine from the list and click **Create Workstations**.
3. In the Workstations Properties panel, specify the attributes for the workstation you are creating, then click **Assign to Domain** and, on the **Select Domain** panel, click **New** to create a new domain.
4. Click **Save** to add the domain to the database and return to the **Select Domain** panel.
5. In the **Select Domain** panel click **OK** to assign the new domain to the workstation you are creating.

To add a new domain definition from the list of distributed workstations, perform the following steps:

1. Click **Administration > Workload Environment Design > List Workstations**.
2. Optionally, enter filtering criteria and click **Display**.
3. From the list of workstations, click **Domains View**.
4. Click **New** and, on the Domain properties panel, specify the properties for the domain.
5. Click **Save** to add the domain to the database or **Cancel** to exit without saving.

Related concepts:

"Domain" on page 34
The domain.

Creating a pool of agents

You can define and schedule dynamic jobs to perform application specific operations, such as database, file transfer, Java, and Web Services operations. You can customize the sample files provided with the product to match the requirements of your environment.

To run these job types, you can use dynamic agents, a workstation type that you create by running the related installation process. The dynamic agents are automatically created and registered at installation time. You can also organize the dynamic agents into groups, called pools or dynamic pools.

For more information about dynamic scheduling, see *Scheduling Workload Dynamically*

To add this kind of workstation definition to the database and to assign it to a domain, perform the following steps.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.



1. From the navigation toolbar, click **Administration > Workload Environment Design > Create Workstations**.
2. Select an engine from the list and click **Create Workstations**.
3. In the Workstations properties panel, specify the attributes for the pool of dynamic agents you are creating. In the **Workstation Type** menu, select **Pool** or **Dynamic Pool**, depending on the set of dynamic agents you are defining.
 - Select **Pool** to define a cluster of dynamic agents with similar hardware or software characteristics to submit jobs to. Then, in the **Member** table that displays the dynamic agents that belong to the Pool, click **Add** to add new dynamic agents and **Remove** to remove unwanted dynamic agents.
 - Select **Dynamic Pool** to define a set of dynamic agents that is dynamically defined based on the resource requirements you specify. Then click **Edit Requirements** to display a panel where you can specify the requirements necessary for running your jobs. All your selections produce an XML file, which is used to select a workstation with the characteristics you require, to run Workload Broker jobs. When you provide the requirements, you specify a list of workstation candidates to be included in the **Dynamic Pool** of dynamic agents and the preferred order in which they must be considered. You also specify the best criteria, which is used to change the workstation (workload balance, CPU utilization, or its use of logical resources).
4. Optionally, you can associate the new pool to a variable table.
5. Specify the Workload Broker hosting the workstation.

Related concepts:

“Workstation” on page 31

Designing your Workload

To create and edit scheduling objects, open the Workload Designer, by doing as follows:



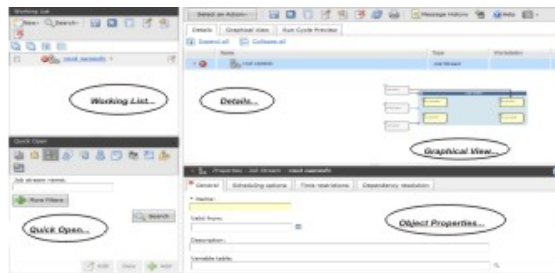
1. From the navigation toolbar, **Administration > Workload Design > Manage Workload Definitions**.
2. In the displayed panel, specify the engine connection you want to use. Only the categories of objects supported by the engine that you selected are available.

You can open multiple occurrences of the Workload Designer. The following scenarios are supported:

- Same user connected to multiple engines
- Multiple users connected to the same engine.

When you open the Workload Designer panel, the following window is displayed, showing several working areas that you can use, depending on the task you want to perform.

Click the links in the picture below to find more information about each pane:



Related concepts:

“Scheduling objects” on page 37

Editing objects from the Working List View

To modify objects from the **Working List** view, perform the following steps:

1. Open the **Search** menu and select the object you want to modify.

2. In the displayed lookup panel, select the required filtering criteria and click **Search**.
3. From the displayed list, select one or multiple objects and click one of the following action buttons:




Create like

To create a new object with the same properties of the selected object.



Edit

To modify the properties of the selected objects. When an object is open in edit mode the edit icon  is displayed on the right of the object.



Unlock

To unlock the selected objects for further actions. When an object is unlocked, it is displayed in read-only mode.



Delete

To delete the selected objects from the database.

All the selected objects are listed in the Working List view. When you select an object from this list, its properties are displayed in the right pane, where you can view or edit them, if you have the required authorization defined in the Tivoli Workload Scheduler security file.

By hovering with the cursor over the icons located on the toolbar, you can see all the actions you can perform on the selected objects.

Related information:

“Working List” on page 214

Editing objects from the Quick Open View

To modify objects from the **Quick Open** view, perform the following steps:

1. Click one of the following icons displayed at the top of the pane to select the category of your search. Available categories depend on the engine connection you selected.

Distributed In a distributed environment



z/OS In a z/OS environment



2. Refine your search by using some filtering criteria. You can filter by object name by entering the name, or part of it, in the text box, or by using wildcards. Optionally, you can further filter your search by choosing more filtering criteria from the drop-down menu.
3. Click **Search**. The list of results is shown, up to a maximum of 250 items. If the list is longer, use the filters to refine your search and reduce the number of results.
4. Select one or more objects to open them in edit mode by clicking the **Edit** button. The open objects are locked in the database until you finish editing and save them.

When an object is open in edit mode, its properties are displayed in the right pane, where you can view and edit them.

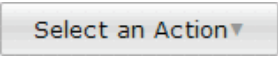
You can drag the objects in the **Quick Open** pane and drop them in the **Graphical View** or you can quickly associate them to the item you are working with, by clicking the **Add** button. For example, you can search for jobs and automatically add them to the job stream you are editing, or add other objects as dependencies, such as resources or prompts.

Related information:

“Quick Open” on page 215

Editing objects from the Details View

To modify object from the **Details** view, perform the following steps:

1. Choose the object you want to modify and either:
 - Select it and click the  button.
 - Right-click it and choose an action from the context menu.
2. Choose one of the displayed actions to act on the selected item. Only actions available for the selected item are displayed in the menus.

When an object is open in edit mode, its properties are displayed in the bottom pane, where you can view and edit them.

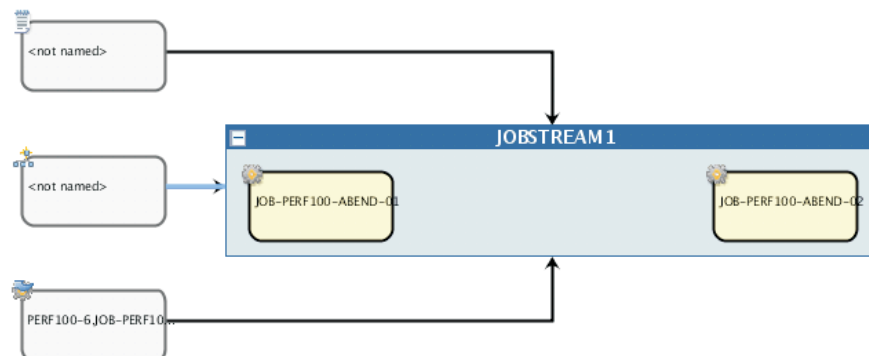
You can also drag the objects contained in the **Quick Open** pane and drop them in the **Details** pane to create job and job stream dependencies and to add jobs to the job stream.

Related information:

“Details view” on page 216

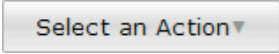
Editing objects from the Graphical View

The **Graphical View** shows the job stream with the jobs it contains and related dependencies.



Note: By opening the menu next to the camera icon on the toolbar, you can see and launch a set of short demos that can help you become familiar with the Graphical View main features.

From the **Graphical View** you can edit an object in any of the following ways:

- Select it and click the  button.
- Right-click it and choose an action from the context menu. Only actions available for the selected item are displayed in the menus.
- Select an object or an arrow and use the buttons on the graphical view toolbar to create or remove dependencies.
- Search for an object in the **Quick Open** pane and click **Add** or drag and drop it on a job or on the job stream. You can use drag and drop to add jobs and dependencies to the job stream. You can also use it to add dependencies to jobs. Consider that, for example, if you drop a job on another job, it is added as an external dependency, even if the job is in the same job stream.

Related information:

“Graphical View” on page 218

Editing object properties

Use the Properties pane, to modify the properties of the selected object.

The properties pane is divided into tabs, which contain options and properties relating to the object currently open.

If you have more than one object open in the **Working List**, the properties displayed in this pane are relating to the object currently selected in the **Details** or **Graphical** view above.

Enter the required information in the mandatory fields. An automatic check allows you to enter only supported characters.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.

Creating job stream definitions

From the Workload Designer panel of the Dynamic Workload Console, you can define job streams, running on both distributed and z/OS environments. Most of them can also be created using composer (on distributed systems) and ISPF (on z/OS systems) interfaces.

Optionally, you can quickly submit the job stream into the current plan to be run immediately. You might want to do this for testing purposes before implementing the job stream in large-scale production. See “Quick submit of jobs and job streams” on page 96.

To create a new job stream definition from the Dynamic Workload Console, perform the following procedure:

1. From the navigation toolbar, click  **Administration > Workload Design > Manage Workload Definitions**

2. Specify an engine name, either distributed or z/OS. The Workload Designer opens. Job types and characteristics vary depending on whether you select a distributed or a z/OS engine.
3. In the Working List pane, select **New > Job Stream**.
4. In the properties panel, specify the attributes for the job stream definition you are creating. For all the details about available fields and options, see the online help by clicking the "?" on the top-right corner.
5. Click **Save** to save the job definition in the database.

Related concepts:

"Job stream" on page 38

"Job" on page 38

Related reference:

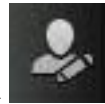
"Status description and mapping for distributed job streams" on page 203

"Status description and mapping for z/OS job streams" on page 205

"Limit the number of objects retrieved by queries" on page 190

Creating job definitions

From the Workload Designer panel of the Dynamic Workload Console, you can define multiple types of jobs, running on both distributed and z/OS environments. Most of them can also be created using composer (on distributed systems) and ISPF (on z/OS systems) interfaces. Some predefined job types are organized into categories including native jobs, and job types with advanced options, which are specific job types that you use to perform operations on external applications. In addition to the existing ones, you can define your own job types and add them to this list. To create a new job definition from the Dynamic Workload Console, perform the following procedure:



1. From the navigation toolbar, click **Administration > Workload Design > Manage Workload Definitions**.
2. Specify an engine name, either distributed or z/OS. The Workload Designer opens. Job types and characteristics vary depending on whether you select a distributed or a z/OS engine.
3. In the Working List pane, select **New > Job Definition**.
4. Select the category and type of job you want to create.
5. In the properties panel, specify the attributes for the job definition you are creating. For all the details about available fields and options, see the online help by clicking the "?" on the top-right corner.
6. Click **Save** to save the job definition in the database.

Optionally, you can quickly submit the job definition to be run. You might want to do this for testing purposes before implementing the job in large-scale production. See "Quick submit of jobs and job streams" on page 96.

Currently, the following job categories and types are available:

Table 8. Job Types

Category	Job Type	Description	job types with advanced options
Native	Windows	Jobs that run on Windows operating systems.	NO
	UNIX	Jobs that run on UNIX platforms.	NO
	Other	Jobs that run on extended agents. Refer to <i>Tivoli Workload Scheduler for Applications User's Guide</i> for information about customized task types for supported vendor acquired applications. Jobs that run on limited fault-tolerant agents for IBM i.	NO
	Executable	Jobs that run script or command with advanced options, such as redirecting standard input and standard output to a file.	YES
	Remote Command	Jobs that can run on remote computers and no agent installation is required. Note: On z/OS systems, it can be created only using the Dynamic Workload Console	YES
	z/OS	Jobs that run the command that you specify in the JCL tab on a JCL system.	YES
	IBM i	Jobs that run a command on IBM i systems.	YES
ERP	SAP Job on XA Workstations	Jobs that run on an SAP extended agent. This includes the three types of SAP R/3 job definitions: <ul style="list-style-type: none"> Standard R/3 job BW Process Chain job BW InfoPackage job 	NO
	SAP Job on Dynamic Workstations	Jobs that run on dynamic agent workstations, pools, dynamic pools, and z-centric agents. The following types of SAP job definition are available: <ul style="list-style-type: none"> Standard R/3 job BW Process Chain job BW InfoPackage job 	NO
	Access Method	Jobs extend job scheduling functions of Tivoli Workload Scheduler to other systems and applications using access methods. The access methods communicate with the external system to launch the job and return the status of the job. The following access methods are available: <ul style="list-style-type: none"> Oracle E-Business Suite PeopleSoft SAP MVS[™] Custom methods 	NO
	PI Channel	Jobs that run SAP Process Integration (PI) Channel jobs to control communication channels between the Process Integrator and a backend SAP R/3 system. Note: Provided with Tivoli Workload Scheduler for Applications 8.6 or later and available only if specifically installed.	YES
Cloud	Workload Broker	Jobs that manage the lifecycle of a dynamic workload broker job. <i>Tivoli Workload Scheduler Scheduling Workload Dynamically</i> for information about how to use dynamic workload broker.	NO
	Provisioning	Jobs that span physical computers, virtual machines, and private and public cloud environments creating an on-demand environment. This job type integrates with IBM SmartCloud Provisioning. Note: On z/OS systems, it can be created only using the Dynamic Workload Console	YES
File Transfer and Coordination	File Transfer	Jobs that run program to transfer files to and from a server reachable using FTP, SSH, or other protocols.	YES
	Shadow Distributed	Jobs that run locally and map other jobs running in remote Tivoli Workload Scheduler distributed environment.	NO
	Shadow z/OS	Jobs that run locally and map other jobs running in remote Tivoli Workload Scheduler for z/OS environment.	NO
Database and Integrations	Database	Jobs that perform queries, SQL statements, and jobs on a number of databases, including custom databases. You can also create and run stored procedures on DB2 [®] , Oracle, and Microsoft SQL Server databases.	YES
	Web Services	Jobs that run a web service.	YES
	MS SQL	Jobs that run a Microsoft SQL Server job.	YES
	J2EE	Jobs that allow Java [™] applications in the same network to send and receive messages from and to a JMS destination.	YES
	Java	Jobs that run a Java class.	YES
Business Analytics	Cognos Reports	Jobs that run IBM Cognos reports, interactive reports, query, and report views. Note: Provided with Tivoli Workload Scheduler for Applications 8.6 or later and available only if specifically installed.	YES
	InfoSphere [®] DataStage [®]	Jobs that run IBM InfoSphere DataStage jobs. Note: Provided with Tivoli Workload Scheduler for Applications 8.6 or later and available only if specifically installed.	YES
OSLC	OSLC Automation	Jobs that invoke any OSLC provider that is implementing the OSLC Automation Specification. Automation resources define automation plans, automation requests and automation results of the software development, test and deployment lifecycle.	YES
	OSLC Provisioning	Jobs that invoke any OSLC provider that is implementing the OSLC Provisioning Specification. Provisioning resources define provisioning plans, provisioning requests and provisioning results of the software development, test and deployment lifecycle.	YES

Note: To create job types with advanced options, ensure you have completed the prerequisite steps described in “Prerequisite steps to create job types with advanced options” on page 91 before performing the following procedure:

For more information about creating the plug-ins to support the definition and running of dynamic job types, see *Developer's Guide: Extending IBM Tivoli Workload Automation*

Related concepts:

"Job" on page 38

"Workstation" on page 31

"Variable table" on page 58

"Run cycle" on page 41

"Dependencies" on page 48

Controlling processing using dependencies

"Customizing your job stream" on page 171

Defining jobs based on the agent type

How to create a job definition for supported agents using the Dynamic Workload Console.

Steps for defining a job for supported agents.

Fault-tolerant agents

To define jobs, follow these steps:

1. From the navigation toolbar, click **Administration>Workload Design>Manage Workload Definitions**.
2. Specify an engine name, either distributed or z/OS. The Workload Designer opens. Job types and characteristics vary depending on whether you select a distributed or a z/OS engine.
3. In the Working List pane, select
 - **Distributed** **New > Job Definition**
 - **z/OS** **New**
4. Select the category and type of job you want to create.
 - For Tivoli Workload Scheduler native jobs, click
 - **Distributed** **Native > Windows, UNIX or Other**.
 - **z/OS** **z/OS job**. See *Scheduling with the Agent for z/OS*.
5. In the properties panel, specify the attributes for the job definition you are creating. For all the details about available fields and options, see the online help by clicking the "?" in the top-right corner.
6. Click **Save** to save the job definition in the database.

Limited fault-tolerant agent

To define jobs, follow these steps:

1. From the navigation toolbar, click **Administration>Workload Design>Manage Workload Definitions**.
2. Specify an engine name, either distributed or z/OS. The Workload Designer opens. Job types and characteristics vary depending on whether you select a distributed or a z/OS engine.
3. In the Working List pane, select
 - **Distributed** **New > Job Definition**
 - **z/OS** **New**
4. Select the category and type of job you want to create.
 - For Tivoli Workload Scheduler native jobs, click
 - **Native > Other**. See *Limited Fault-tolerant Agent for IBMi*.

5. In the properties panel, specify the attributes for the job definition you are creating. For all the details about available fields and options, see the online help by clicking the "?" in the top-right corner.
6. Click **Save** to save the job definition in the database.

Agent for z/OS

To define jobs, follow these steps:

1. From the navigation toolbar, click **Administration>Workload Design>Manage Workload Definitions**.
2. Specify an engine name, either distributed or z/OS. The Workload Designer opens. Job types and characteristics vary depending on whether you select a distributed or a z/OS engine.
3. In the Working List pane, select
 - **Distributed** **New > Job Definition**
 - **z/OS** **New**
4. Select the category and type of job you want to create.
 - For Tivoli Workload Scheduler native jobs, click
 - **Native > z/OS**. See *Scheduling with the Agent for z/OS*.
5. In the properties panel, specify the attributes for the job definition you are creating. For all the details about available fields and options, see the online help by clicking the "?" in the top-right corner.
6. Click **Save** to save the job definition in the database.

Remote engines

To define shadow jobs, follow these steps:

1. From the navigation toolbar, click **Administration>Workload Design>Manage Workload Definitions**.
2. Specify an engine name, either distributed or z/OS. The Workload Designer opens. Job types and characteristics vary depending on whether you select a distributed or a z/OS engine.
3. In the Working List pane, select
 - For Tivoli Workload Scheduler native jobs,
 - **Distributed** **New > Job Definition**.
 - **z/OS** **New**.
4. Select the category and type of job you want to create.
 - For Tivoli Workload Scheduler jobs,
 - For Tivoli Workload Scheduler native jobs,
 - **Distributed** **File Transfer and Coordination > Shadow Distributed or Shadow z/OS**. See *Tivoli Workload Scheduler User's Guide and Reference*, section about Job definition - Shadow jobs.
 - **z/OS** **Shadow Distributed or Shadow z/OS**. See *Tivoli Workload Scheduler for z/OS Managing the Workload*, section about Specifying cross dependencies and shadow jobs.
5. In the properties panel, specify the attributes for the job definition you are creating. For all the details about available fields and options, see the online help by clicking the "?" in the top-right corner.
6. Click **Save** to save the job definition in the database.

Extended agents

To define jobs, follow these steps:

1. From the navigation toolbar, click **Administration>Workload Design>Manage Workload Definitions**.
2. Specify an engine name, either distributed or z/OS. The Workload Designer opens. Job types and characteristics vary depending on whether you select a distributed or a z/OS engine.
3. In the Working List pane, select **New > Job Definition**.
4. Select the category and type of job you want to create.
 - For SAP jobs, **ERP > SAP Job on XA Workstations**. See *Tivoli Workload Scheduler for Applications User's Guide*, section about Creating SAP Standard R/3 jobs from the Dynamic Workload Console.
 -
 - For Oracle E-Business Suite, and PeopleSoft, **Native > Other**.
 - For z/OS, **Native > z/OS**.
5. In the properties panel, specify the attributes for the job definition you are creating. For all the details about available fields and options, see the online help by clicking the "?" in the top-right corner.
6. Click **Save** to save the job definition in the database.

Dynamic agent and Tivoli Workload Scheduler for z/OS agents

The steps described here for defining a job are common to all access methods.

Follow these steps:



1. From the navigation toolbar, click **Administration > Workload Design > Manage Workload Definitions**.
2. Specify an engine name, either distributed or z/OS. The Workload Designer opens. Job types and characteristics vary depending on whether you select a distributed or a z/OS engine.
3. In the Working List pane, select
 - **Distributed** **New > Job Definition**
 - **z/OS** **New**
4. Select the category and type of job you want to create.
 - For SAP R/3 jobs, **ERP**,
 - **Distributed** **SAP Job on Dynamic Workstations**. See *Tivoli Workload Scheduler for Applications User's Guide*, section about Creating SAP Standard R/3 jobs from the Dynamic Workload Console.
 - **z/OS** **SAP**. See *Tivoli Workload Scheduler for Applications User's Guide*, section about Creating SAP Standard R/3 jobs from the Dynamic Workload Console.
 - **SAP PI Channel**, see *Tivoli Workload Scheduler for Applications User's Guide*, section about Defining a Tivoli Workload Scheduler job that runs an SAP PI Channel job.
 - For Tivoli Workload Scheduler native job types, **Native**
 - **Executable**, see *Tivoli Workload Scheduler User's Guide and Reference*, section about Job definition - Executable jobs
 - **IBM i**, see "Scheduling jobs on IBM i" in *Limited Fault-Tolerant Agent IBM i*.

- **z/OS**, see *Tivoli Workload Scheduler for z/OS Scheduling End-to-end with z-centric Capabilities*.
 - **Remote command**, see *Tivoli Workload Scheduler User's Guide and Reference*, section about Job definition -remote command jobs.
 - For **Database and Integrations** job types with advanced options, **Database and Integrations**,
 - **Database**, see *Tivoli Workload Scheduler User's Guide and Reference*, section about Job definition - Database jobs.
 - **J2EE**, see *Tivoli Workload Scheduler User's Guide and Reference*, section about Job definition - J2EE jobs.
 - **Java**, see *Tivoli Workload Scheduler User's Guide and Reference*, section about Job definition - Java jobs.
 - **MS SQL**, see *Tivoli Workload Scheduler User's Guide and Reference*, section about Creating job types with advanced options.
 - **Web Services**, see *Tivoli Workload Scheduler User's Guide and Reference*, section about Job definition - Web services jobs.
 - For **Business Analytics** job types with advanced options, **Business Analytics**,
 - InfoSphere DataStage, see *Tivoli Workload Scheduler for Applications User's Guide*, section about Defining a Tivoli Workload Scheduler job to run an InfoSphere DataStage job.
 - Cognos, see *Tivoli Workload Scheduler for Applications User's Guide*, section about Defining jobs by using composer.
 - For **Cloud** environments, **Cloud > IBM SmartCloud Provisioning**, see *Tivoli Workload Scheduler User's Guide and Reference*, section about Job definition - IBM SmartCloud Provisioning jobs
 - For **Access Method** job types, Oracle E-Business Suite, and PeopleSoft, **ERP > Access Method**.
 - For **File Transfer and Coordination > File Transfer**, see *Tivoli Workload Scheduler User's Guide and Reference*, section about Job definition - File transfer jobs.
5. In the properties panel, specify the attributes for the job definition you are creating. For all the details about available fields and options, see the online help by clicking the "?" in the top-right corner.
 6. Click **Save** to save the job definition in the database.

Note: The access method for SAP R/3 provides supplementary features if you use the alternative steps described in *Tivoli Workload Scheduler for Applications User's Guide*, section about Create an SAP job and associate it to a Tivoli Workload Scheduler job or Creating SAP Standard R/3 jobs from the Dynamic Workload Console. You can create native SAP R/3 Standard jobs on a remoteSAP system directly from the Dynamic Workload Console.

Prerequisite steps to create Provisioning jobs

To create a Provisioning job definition, you must first complete the prerequisite steps listed below.

For more information about creating Provisioning job definitions, see the relevant chapter about Job definition - SmartCloud Provisioning jobs in the *Tivoli Workload Scheduler User's Guide and Reference*.

1. Obtain the SmartCloud HTTP server certificate and save it in a directory that you will later use. Provisioning administrator can provide the certificate, or, with Firefox browser, for example, you can retrieve the certificate performing the following steps:
 - a. Log in to Provisioning server providing Provisioning credentials.
 - b. Download the certificate by clicking in the browser toolbar:
Tools>Options>Advanced>Encryption>View Certificates
 - c. Select **IBM> IBM Websphere Cloudburst Appliance** and click **Export**. A file, named IBMWebSphereCloudBurstAppliance.crt (X509 Certificate - PEM) is created.
2. Browse to the directory where a JRE is installed, for example: C:\Program Files\IBM\TWS\javaExt\JRE\JRE
3. Create a new truststore by launching the following command: `keytool -genkeypair -alias certificatekey -keyalg RSA -validity 7 -keystore keystore.jks`,
where, *keystore.jks* is the file path to the keystore.
4. Add IBM SmartCloud certificate to the truststore by launching the following command: `keytool -import -file certificate_directory\IBMWebSphereCloudBurstAppliance.crt -alias scp -keystore trustore_directory\keystore.jks`,
5. Open the *TWA_HOME\TWS\ITA\cpa\config\JobManager.ini* file, and locate JavaJobLauncher section, JVMOptions row.
6. Add the following instructions to the row: "-Djavax.net.ssl.trustStore= *DIRECTORY_TRUSTSTORE*/keystore.jks
-Djavax.net.ssl.trustStorePassword=*TRUSTSTORE_PASSWORD*" For example:
JVMOptions = -Djavax.net.ssl.trustStore=C:/myUtils/keystore.jks
-Djavax.net.ssl.trustStorePassword=passw0rd
7. To complete the procedure, stop and restart the agent.

Prerequisite steps to create OSLC Automation and OSLC Provisioning jobs

To create an OSLC Automation or OSLC Provisioning job definition, you must first complete the prerequisite steps listed hereafter.

Note: Before performing the following procedure, ensure that you installed the Jazz for Service Management Registry Services from the Installation Manager.

1. Obtain the Registry Services server certificate and save it in a directory that you will later use. Registry Services administrator can provide the certificate, or, with Firefox browser, for example, you can retrieve the certificate performing the following steps:
 - a. Log in to a Registry Services (for example, <https://hostname:16311/oslc/pr>)
 - b. Download the certificate by clicking in the browser toolbar:
Tools>Options>Advanced>Encryption>View Certificates
 - c. Select **IBM>Registry_Services_hostname:port** and click **Export**. A file is created with the name that you specify, for example myserver:16311.
2. Browse to the directory where a JRE is installed, for example: C:\Program Files\IBM\TWA_<TWS_user>\TWS\JavaExt\jre\jre\bin
3. Create a new truststore by launching the following command: `keytool -genkeypair -alias certificatekey -keyalg RSA -validity 7 -keystore trustore_directory\keystore.jks`

Note: Ensure that the *trustore_directory* is not created in the javaExt\JRE path.

4. Add the IBM Registry Services certificate to the truststore by launching the following command: `keytool -import -file certificate_directory\certificate_name -alias oslc -keystore trustore_directory\keystore.jks`
5. Open the *TWA_HOME\TWS\ITA\cpa\config\JobManager.ini* file, and locate JavaJobLauncher section, JVMOptions row.
6. Add the following instructions to the row:
"-Djavax.net.ssl.trustStore=DIRECTORY_TRUSTSTORE/keystore.jks
-Djavax.net.ssl.trustStorePassword=TRUSTSTORE_PASSWORD". For example:
JVMOptions = -Djavax.net.ssl.trustStore=C:/myUtils/keystore.jks
-Djavax.net.ssl.trustStorePassword=passw0rd
7. Stop and restart the agent.
8. Create the *OSLCAutomation.properties* and *OSLCProvisioning.properties* files, respectively for the OSLC Automation and OSLC Provisioning jobs, and locate them in *TWA_Home/TWS/JavaExt/cfg/*.
Specify the service provider catalogs (or Registry Services) that you will later use to create the job in the following format:
ServiceProviderCatalogName=RegistryServicesURI
9. On the master domain manager, extract the security certificate from the keystore and add it to the Jazz for Service Management truststore. The file paths and names are as follows:

Master domain manager keystore

<TWA_Home>/WAS/TWSPProfile/etc/TWSServerKeyFile.jks

Jazz for Service Management truststore

<JazzSM_Home>/profile/config/cells/JazzSMNode01Cell/nodes/
JazzSMNode01Cell/trust.p12

10. On the workstation where you installed Jazz for Service Management, extract the security certificate from the keystore and add it to the master domain manager truststore. The file paths and names are as follows:

Jazz for Service Management keystore

<JazzSM_Home>/profile/config/cells/JazzSMNode01Cell/nodes/
JazzSMNode01Cell/key.p12

Master domain manager truststore

<TWA_Home>/WAS/TWSPProfile/etc/TWSServerTrustFile.jks

11. Close and restart the WebSphere Application Server on the master domain manager and on Jazz for Service Management.

Prerequisite steps to create job types with advanced options

Most job types include additional operations which vary depending on the job type to help you define and schedule the job correctly. For example, you can use these job-specific options to check for the connection to the database when defining a database job or to retrieve the list of available operations for a web service when defining a web service job. If you try to perform job-specific options on a dynamic agent, version 8.5.1, fix pack 1, an error message is returned because the options are not supported, but the job can be scheduled correctly.

In the distributed environment, you define job types with advanced options using the Dynamic Workload Console connected to a distributed engine or the **composer** command.

For more information about defining job types with advanced options using the **composer** command, see *User's Guide and Reference*.

In the z/OS environment, you define job types with advanced options using the Dynamic Workload Console connected to a z/OS engine or the JOBRECE statement.

For more information about defining job types with advanced options using the JOBRECE statement, see the section about defining a job in the JOBLIB data set in this guide.

In both environments, you can use the related configuration files to define options for some job types with advanced options.

For more information, see the section about configuring to schedule job types with advanced options in the *Administration Guide* and in this guide.

In addition, you can create custom plug-ins to implement your own job types with advanced options for applications not supported by Tivoli Workload Scheduler.

For more information about how to create custom plug-ins, see the *Tivoli Workload Automation Developer's Guide*

You can schedule job types with advanced options only on dynamic agents, pools, and dynamic pools.

This procedure describes how to create a file transfer job in both the distributed and z/OS environments. The procedure for creating other job types is similar, but each job type contains job-specific options. For more information about each job type, see the Dynamic Workload Console online help. To create a file transfer job, perform the following steps:

1. Install a number of dynamic agents and add the Java run time

To install dynamic agents, run the installation program. You can install the dynamic agent during the full installation of Tivoli Workload Scheduler or in a stand-alone installation of just the agent. During the installation, you have the option of adding the Java run time to run job types with advanced options, both those types supplied with the product and the additional types implemented through the custom plug-ins.

Follow the installation wizard to complete the installation.

See the section about installation options in *Planning and Installation Guide* for descriptions of the installation parameters and options.

2. Organize the dynamic agents in pools and dynamic pools.

Pools and dynamic pools help you organize the environment based on the availability of your workstations and the requirements of the jobs you plan to run.



a. From the navigation toolbar, click **Administration > Workload Environment Design > Create Workstations**

b. Select a distributed or z/OS engine. The workstations you can create vary depending on the engine type you select.

c. Select the workstation type you want to create.

- To create a pool, define the dynamic agents you want to add to the pool and the workload broker workstation where the pool is hosted.

- To create a dynamic pool, specify the requirements that each dynamic agent must meet to be added to the dynamic pool.

3. Grant the required authorization for defining job types with advanced options.

The Tivoli Workload Scheduler administrator has to grant specific authorizations in the security file to allow the operators to create job types with advanced options.

- In the distributed environment, perform the following steps:
 - a. Navigate to the *TWA_home*/TWS directory from where the **dumpsec** and **makesec** commands must be run.
 - b. Run the **dumpsec** command to decrypt the current security file into an editable configuration file.
For more information, see the section about **dumpsec** in *Tivoli Workload Scheduler Administration*.
 - c. Add display and run access to the workstation, as follows:
 - If the operation is performed on the Tivoli Workload Scheduler Connector, display and run access is required on the CPU corresponding to the workstation where the job is created.
 - If the operation is performed on the workstation where the job runs, display access is required on the workload broker workstation.
 For more information, see the section about configuring the security file in *Tivoli Workload Scheduler Administration*.
 - d. Close any open **conman** user interfaces using the **exit** command.
 - e. Stop any connectors on systems running Windows operating systems.
 - f. Run the **makesec** command to encrypt the security file and apply the modifications.
For more information, see the section about **makesec** in *Tivoli Workload Scheduler Administration*.
 - g. If you are using local security, the file is immediately available on the workstation where it has been updated.
 - 1) If you are using a backup master domain manager, copy the file to it.
 - 2) Distribute the centralized file manually to all fault-tolerant agents in the network (not standard, extended, or broker agents), and store it in the *TWA_home*/TWS directory.
 - 3) Run **JnextPlan** to distribute the Symphony file that corresponds to the new security file.

- In the z/OS environment, perform the following steps:
 - a. Define the fixed resource that owns the subresource and give universal read access to it:
`RDEFINE (CLASS_NAME) FIXED_RESOURCE UACC(READ)`
 - b. Give user *USER_ID* update access to the *FIXED_RESOURCE* fixed resource:
`PERMIT FIXED_RESOURCE ID(USER_ID) ACCESS(UPDATE) CLASS(CLASS_NAME)`
 - c. Define a RACF resource, *JSORACF_RESOURCE*, to RACF and give universal read access to it:
`RDEFINE (OPCCCLASS) JSORACF_RESOURCE UACC(READ)`

JSO is the 3-character code that RACF uses for JS.OWNER.

- d. Give user *USER_ID* update access to *JSORACF_RESOURCE*:
`PERMIT JSORACF_RESOURCE ID(USER_ID) ACCESS(UPDATE) CLASS(CLASS_NAME)`

4. Define the job types with advanced options as described in “Creating job definitions” on page 84.

In the distributed environment, you can define job types with advanced options also using the **composer** command.

For more information, see the section about job definition in the *Tivoli Workload Scheduler User's Guide and Reference*

In the z/OS environment, you can define job types with advanced options also using the **JOBREC** command.

For more information, see **JOBREC**.

Related concepts:

“Job” on page 38

“Variable table” on page 58

“Run cycle” on page 41

“Dependencies” on page 48

Controlling processing using dependencies

“Customizing your job stream” on page 171

Specifying the parameter format for date, time, and time stamp

When defining reports either using **composer** or the Dynamic Workload Console, specify parameters of type date, time, and time stamp, using a specific syntax.

The following table shows the syntax you must use when defining reports containing date, time, and time stamp formats as parameters.

Table 9. Examples to use for parameters of date, time, and time stamp formats

Prompt type	Cognos parameter format	Cognos parameter format examples		
		Single value	List of values	Interval values
Date	CCYY-MM-DD	2012-02-03	2012-02-03-Value:2012-03-14	Between 2012-02-03 and 2012-04-15
Time	hh:mm:ss	01:00:00	01:00:00-Value:01:01:01	Between 01:00:00 and 23:59:30
Time Stamp	CCYY-MM-DDThh:mm:ss or CCYY-MM-DD hh:mm:ss	2012-02-03 15:05:00	2012-02-03 15:05:00-Value:2012-02-03T16:01:00-Value:2012-02-03T16:00:00	Between 2012-02-03 15:05:00 and 2012-04-15T16:00:00

Note: You must specify the parameter format exactly as they are shown in the table respecting lower case and upper case formats.

Adding a job to a job stream

To perform this task, you must have a connection with a Tivoli Workload Scheduler engine up and running. You must also be connected to the database where the object definitions are stored.

To edit an object in the database you can use the different views available in the Workload Designer window.

1. You can open an existing job stream in either of the following ways:
 - Perform the following steps:



- From the navigation toolbar, click **Administration > Workload Design > List Workload Definitions**.
- Specify the engine connection and filtering criteria to retrieve the job stream to which you want to add the job.
- Select the job stream and click **Edit**.

The Workload Designer panel opens in a new window displaying the selected job stream open in edit mode.

- Perform the following steps:



- From the navigation toolbar, click **Administration > Workload Design > Manage Workload Definitions**.
- Specify the engine connection and the Workload Designer panel opens in a new window.
- Open the job stream to which you want to add a job. To find the job stream you need, use either of the following ways:
 - Search for it using the **Search** menu in the **Working List** pane

- Select the job stream icon  and launch a search (optionally filtered) in the **Quick Open** pane.

- You can now work with the job stream by using either the **Graphical** or the **Details** view. Select the tab for the view you want in the right part of the panel.

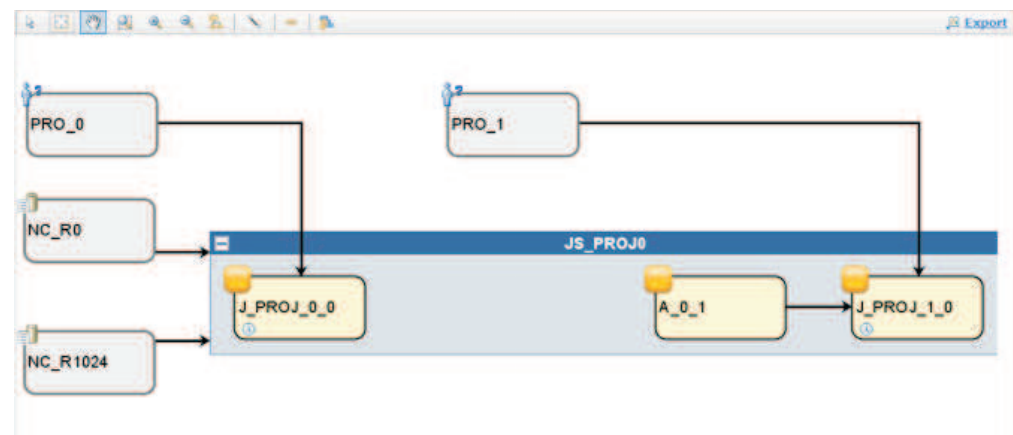
Related concepts:

"Job" on page 38

"Job stream" on page 38

Adding a job from the Graphical View

You can add a job to the job stream using the **Graphical View**.



From the **Graphical View** you can add a job to a job stream in either of the following ways:

- Perform the following steps:
 - Right-click inside the job stream area, and select **Add Jobs** from the menu

2. Search for the required job by using the displayed Search panel and add it to the job stream.
- Search for the job in the **Quick Open** pane and click **Add** or drag and drop it on the job stream. Consider that if you drop the job on another job contained in the job stream, it is added as an external dependency, even if the job is in the same job stream.
 - You can then edit the properties of the newly added job, optionally specifying its scheduling options and time restrictions. For example, here you can set a job as **Critical** if the job is mission critical and must be processed accordingly.

Adding a job from the Details View

If you use the **Details** view, you see a tree-table view of the job stream with all its jobs, dependencies, and run cycles.

Name	Type	Workstation	
FNAL	Job Stream	NC122071	
Dependencies			
FNAL SWITCHPLAN	External Job Dependency	NC122071	✖
Run Cycles			
RULE1	Run Cycle		✖
Jobs			
STARTAPPSERVER	Job	NC122071	✖
MAKEPLAN	Job	NC122071	✖
SWITCHPLAN	Job	NC122071	✖
CREATEPOSTREPORTS	Job	NC122071	✖
UPDATESTATS	Job	NC122071	✖

From the **Details** view you can add a job to a job stream in either of the following ways:

- Perform the following steps:
 1. Right-click the job stream row and select **Add Jobs** from the menu.
 2. Search for the required job by using the Search panel, and add it to the job stream.
- Search for the job in the **Quick Open** pane and click **Add** or drag and drop it on the job stream. Consider that if you drop the job on another job contained in the job stream, it is added as an external dependency, even if the job is in the same job stream.
- You can then edit the properties of the newly added job, optionally specifying its scheduling options and time restrictions. For example, here you can set a job as **Critical** if the job is mission critical and must be processed accordingly.

If you want to remove the job click the remove icon  at the end of the job row.

Quick submit of jobs and job streams

Quickly submit a job or job stream from the Workload Designer.


To perform this task, you must have a connection with a Tivoli Workload Scheduler engine up and running. You must also be connected to the database where the object definitions are stored.

You can quickly submit a job or job stream defined in the database into the current plan and run it immediately from the Workload Designer. You might want to do this for testing purposes before implementing the job or job stream in large-scale production to verify, for example, if the job or job stream completes successfully or how long the job takes to complete.

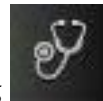
To manage objects in the database, open the Workload Designer window from the portfolio by clicking **Administration > Workload Design > Manage Workload Definitions**.

Submitting a job stream

To submit a job stream from the Workload Designer:

1. Open the job stream that you want to submit to be run. To find the job stream, do either of the following actions:
 - Search for it using the **Search** menu in the **Working List**.
 - Select the job stream icon  and launch a search (optionally filtered) in the **Quick Open** pane.
2. From the **Details** view, click **Select an Action > Submit Job Stream into Current Plan**.


The job stream is submitted to the current plan to be run. You can now monitor



the status of the job stream by clicking **System Status and Health > Workload Monitoring > Monitor Job Streams** from the navigation bar of the Dynamic Workload Console.

Submitting a job

To submit a job from the Workload Designer:

1. Create a new job definition or open the job definition that you want to submit to be run. To find the job definition, do either of the following actions:
 - Search for it using the **Search** menu in the **Working List**.
 - Select the job icon  and launch a search (optionally filtered) in the **Quick Open** pane.
2. From the **Details** view, click **Select an Action > Submit Job into Current Plan**.
3. You are prompted to enter the following optional information:

Job stream name

Type the name of the job stream into which you want to submit the job to run, or browse and select from a list of available job streams defined in the database. If you do not specify a job stream, the job is automatically submitted into the default JOBS job stream.

Job stream workstation name

The name of the workstation where the job stream is defined. If you do not specify a (job stream) workstation name, then the name used is that of the workstation where the JOBS job stream is defined.

Job alias

A name assigned to the job so that it can be easily recognized when monitoring the job run. If you do not specify an alias, it is automatically generated using the first six characters of the job **Name** followed by 10 numerical characters.

The job is submitted to the current plan to be run. You can now monitor the status



of the job by clicking **System Status and Health > Workload Monitoring > Monitor Jobs** from the navigation bar of the Dynamic Workload Console.

Adding and removing dependencies

To perform this task, you must have a connection with a Tivoli Workload Scheduler engine up and running. You must also be connected to the database where the object definitions are stored.


To manage objects in the database, open the Workload Designer window from the




navigation bar by clicking **Administration > Workload Design > Manage Workload Definitions**.

Adding a dependency

To add a dependency, from the Workload Designer, perform the following steps:

1. Open the job stream to which you want to add a dependency or the job stream containing the job to which you want to add a dependency. To find the job stream, do either of the following actions:
 - Search for it using the **Search** menu in the **Working List**
 - Select the job stream icon  and launch a search (optionally filtered) in the **Quick Open** pane.
2. You can now work on the job stream by using either the **Details** or the **Graphical** view. Select the tab for the view you want.
 - From the **Details** view, do either of the following actions:
 - by using the menus, do the following steps:
 - a. Right-click the job or job stream and select **Add Dependencies** from the context menu or select the object and choose an option from the **Select an Action** menu.
 - b. From the context menu, select the type of dependency you want to add.
 - c. From the **Search** panel, search for the required object and add it to the job or job stream.
 - by using drag and drop, do the following steps:
 - a. Select an object from the **Quick Open** pane.
 - b. Drag and drop it on the object that must depend on it.
 - From the **Graphical** view, do either of the following actions:
 - by using the menus, do the following steps:
 - a. Right-click the job or job stream and select **Add Dependencies** from the context menu or select the object and choose an option from the **Select an Action** menu.

- b. From the context menu, select the type of dependency you want to add.
- c. From the **Search** panel, search for the required object and add it to the job or job stream.
- by using the toolbar, do the following steps:
 - a. Select the create dependency icon .
 - b. Draw a line from the dependency to the job or job stream that must depend on it. You can use this tool to create dependencies only between objects displayed in the view.
- by using drag and drop, do the following steps:
 - a. Select an object from the **Quick Open** pane.
 - b. Drag and drop it on the object that must depend on it.



Related concepts:

“Dependencies” on page 48

Controlling processing using dependencies

Removing a dependency

From the Workload Designer, you can remove a dependency in either of the following ways:

- From the **Details** view, click the remove icon  at the end of the dependency row.
- From the **Graphical View**, select the dependency by clicking the node (if it is an external dependency) or by right-clicking the arrow. When the arrow you want to remove is highlighted, click the remove icon  on the toolbar or from the context menu. If an object has multiple dependencies, click it again until the arrow you want to remove is highlighted.

Related concepts:

“Dependencies” on page 48

Controlling processing using dependencies

Creating cross dependencies

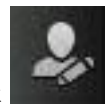
To add a cross dependency to a local job on a job that is defined on a remote engine, you must define a normal dependency for your local job on a shadow job that:

- Points to the remote job on which you want to create the cross dependency
- Is defined on a local workstation of remote engine type, that points to the engine where the remote job is defined.

To define the cross dependency, perform the following steps:

1. Define a remote engine workstation

On a remote engine workstation you can run only shadow jobs. As a prerequisite for creating a remote engine workstation, you must ensure that an HTTP or HTTPS destination to communicate with the remote engine exists. The name of this destination must be specified in the remote engine workstation definition.



- a. From the navigation toolbar, click **Administration > Workload Environment Design > Create Workstations**.
- b. Select an engine from the list and click **Create Workstations**.

- c. Specify the workstation attributes, selecting **Remote engine** as **Workstation Type**. For more details about the options to select, see the online help by clicking the question mark located at the top right corner of the panel.

2. Define a shadow job

It runs on the remote engine workstation and must contain the key values to identify the remote job.



- a. Click **Administration > Workload Design > Manage Workload Definitions** to define a job of type **Shadow Job**.
- b. Select an engine from the list and click **Go**. You can select either a distributed engine (to create a shadow job as an independent object) or on a z/OS engine (to create a shadow job within an existing job stream.)
- c. The Workload Designer opens. In the Working List panel, create the new job.

z/OS

New > File Transfer and Coordination

Distributed

New > Job Definition > File Transfer and Coordination

- d. Select either **Shadow Distributed** or **Shadow z/OS**, depending on the type of remote engine on which the remote job is scheduled to run. For more details about the attributes to select, see the online help by clicking the question mark located at the top right corner of the panel.

Shadow jobs can be added to the plan by the plan creation process or dynamically at run time. The shadow job scheduled time is used to identify the job instance in the remote engine plan.

3. Add the shadow job as a dependency of the local job

The dependency can be internal (distributed only) or external. In the **Details** view, right-click the local job and use the pop-up menu to add the shadow job as a dependency.

As soon as the shadow job status satisfies the dependency rule, the dependency of the local job on the shadow job is resolved, and the cross dependency for the local job on the remote job is also resolved.

For more information about cross dependencies, see the sections about defining and managing cross dependencies in *Tivoli Workload Scheduler User's Guide and Reference* and in *Tivoli Workload Scheduler for z/OS Managing the Workload*

Related concepts:

"Cross dependencies" on page 55

Creating and managing run cycle groups and their run cycles

Why using run cycle groups

By using run cycle groups, you can benefit from the following advantages:

A run cycle group is a distinct database object

It is defined by itself and can be matched with one or more job streams. It is not defined as part of a specific job stream like single run cycles.

The same run cycle group can be used on different job streams

This improves the overall usability of the run cycles, because you can specify the same run cycle group in multiple job streams, avoiding the need to have multiple run cycle definitions for the same scheduling rules.

Run cycle groups enhance the use of exclusive run cycles

Exclusive (or negative) run cycles are used to generate negative occurrences, which identify the days when a job stream would normally be scheduled but is not required. The sum of the exclusive run cycles are subtracted from the inclusive ones. A negative occurrence always cancels any matching positive occurrences and you can specify a negative occurrence only if the positive equivalent already exists. An exact matching of the days, as well as any time restrictions, is required between the exclusive and inclusive run cycles for the cancellation to occur. Run cycle groups add much flexibility by allowing users to apply exclusive run cycles to a subset of the positive ones rather than to all of them. Group your run cycles into *subsets* so that the exclusive run cycles can be applied only to the positive occurrences generated by the run cycles belonging to the same set.

Run cycles must be organized into *subsets* within a run cycle group. The subsets are always in a logical **OR** relationship with each other. The result of the run cycle group is always a date or set of dates; it cannot be negative.

For example, you might want your job stream to run every day of the month except the last day of the month. But, you also want the it to be scheduled on the last day of the year (the last day of December). You can define a run cycle group using subsets, as follows:

Subset 1

- **Run cycle 1** - Inclusive run cycle every day of the month
- **Run cycle 2** - Exclusive run cycle on the last day of the month

Subset 2

- **Run cycle 3** - Inclusive run cycle on December 31st

where, run cycle 2 cancels the last day of each month in Subset 1, while run cycle 3 generates December 31st as a separate date and therefore you can schedule the job stream on Dec 31st.

Run cycle groups allow the use of a logical AND between individual run cycles in the subset

By default, the run cycles within a subset are in a logical **OR** relationship but you can change this to a logical **AND**, if the run cycle group result is a positive date or set of dates (Inclusive). For each run cycle, you can specify either operator (**AND**, **OR**), obtaining the following behavior:

1. All the run cycles of the group that are in **AND** relationship are calculated first. The result of this calculation is a date or a set of dates.
2. Then, all the run cycles in an **OR** relationship are added to the result of the previous step.

A similar behavior is applied to inclusive and exclusive run cycles to determine the final date or set of dates of a group.

Inclusive (A)

Rule-based run cycle. Select days when the job stream is to be run if they belong to all A types of the set of run cycles.

Exclusive (D)

Exclusion rule-based run cycle. Select days when the job stream is NOT to be run if they belong to all D types of the set of run cycles.

For example, you can add two conditions together:

Run on Wednesday "AND" the 8th workday of the month.

In this way, the only scheduled dates are any 8th work day of the month that falls on a Wednesday.

Full compatibility with *traditional* run cycles

The *traditional* run cycles specified in the job stream definition can reference run cycle groups, with the possibility to specify shift or offsets on them (as with periods for z/OS or calendars for distributed systems).

A set of dates (interval starts) is created automatically either at run cycle level directly (inclusively or exclusively with offsets, or in the rule. This is a two-step process with run cycles:

1. Define the key "business event", such as, Month End, using run cycles and free day rules
2. Define rules that use the dates of the "business event" as the intervals against which the other batch run can be scheduled relative to.

For example, you have a *Month End process* that runs on the Last Friday of a month, but that moves forward to the next working day, except in December when it runs on the 3rd Friday of the month. This scheduling rule can be defined with a few rules, run cycles, and free day rules.

Two working days before Month End you need to run a pre-validation process to allow problems to be addressed before the run. You cannot choose the last Wednesday of the month, because in some months this might occur after the last Friday. Similarly, if the last Friday was a free day, the last Wednesday will not be 2 working days before it, because the Free Day rule applies ONLY to the day the rule falls on, it cannot look at anything else.

Many other batch runs might also need to be run on a certain number of days before or after the Month End, but the same restrictions apply.

You can now define work to run relative to something defined by a combination of run cycles and free day rules.

Use of calendars with run cycles within a run cycle group

Optionally, you can specify more than one calendar to calculate the working and non-working days definition for a run cycle. The primary calendar is used to calculate which working days are valid, and a secondary calendar is used to calculate specific

non-working dates. If the dates calculated according to the secondary calendar match with the working days in the primary calendar, the job is scheduled; if they do not match, the job is not scheduled.

For example, a global company that runs workload in the United States for many other countries needs many calendar combinations to ensure that the batch jobs only run on a day that is a working day both in the United States and the other country. The calendar can be defined at job stream level and, if not specified, a default calendar is used. However, the calendar at run cycle level, whenever defined, can be used as secondary calendar and the job stream (or default) calendar can be used as the primary calendar.

For example, Primary calendar can be *WORKDAYS*, which is defined as MONDAY to FRIDAY excluding US holiday dates. You might also need to calculate the job runs based on calendar *HKWORK*, which is defined as Monday to Friday excluding Hong Kong holiday dates. The job might have several schedules:

- Run on working days, but not the last working day and not Mondays
- Run on Mondays, but not on the last working day
- Run on the last working day

Because each schedule is calculated against the *WORKHK* calendar it is also checked against the *WORKDAYS* calendar to ensure that it is scheduled on a US working day.

The use of time restrictions with run cycle groups

You can specify time constraints to define the time when processing must start or the time after which processing must no longer start. To do this, you can associate *time restrictions* to job, job streams, run cycles, and run cycle groups. When you define a time restriction, you basically obtain a *time*. Because you can associate time restrictions to multiple objects, the following hierarchy shows the order by which the different time restrictions are taken into consideration to actually define when to start the processing:

1. Time restriction defined in the run cycle into the job stream
2. Time restriction defined in the job stream
3. Time restriction defined in the run cycle contained in the run cycle group associated to the job stream.
4. Time restriction defined in the run cycle group associated to the job stream.
5. Start of Day

This means that:

Time restrictions in the job stream

Override and take precedence *over any other time restrictions* defined in the run cycles or run cycle groups associated to the job stream.

No time restrictions in the job stream nor in the run cycle group

The group generates only a date that is the *Start Of Day*. If offsets and free day rules are to be calculated, the calculation always starts from the *Start Of Day*.

Time restrictions in the run cycle group (not in the job stream)

Time restrictions (and possible offset) are calculated starting from the *Start Of Day* and the resulting date and time indicate the start of processing.

Examples

Table 10. Scenario 1. No time restriction in the run cycle group

Run cycle group	Scheduled date	Earliest Start
Run cycle group	10/24	10/24
Run cycle group with offset (+ 3 days)	10/27 (Saturday)	10/27/ (Saturday)
Run cycle group with free day rule	10/29/ (Monday)	0/29/ (Monday)
Run cycle in the job stream with time restrictions		
Run cycle in the job stream with + 4 working days shift	11/02 (Friday)	11/02 (Friday)
Run cycle in the job stream with free day rule	11/02 (Friday)	11/02 (Friday)
Run cycle in the job stream with earliest start +1 1pm	11/02 (Friday)	11/03 (Saturday) 1pm
Run cycle in the job stream without time restrictions		
Run cycle in the job stream with + 4 working days shift	11/02 (Friday)	11/02 (Friday) Start of Day
Run cycle in the job stream with free day rule	11/02 (Friday)	11/02 (Friday) Start of Day

Table 11. Scenario 2. Time restriction in the run cycle group without offset

Run cycle group	Scheduled date	Earliest Start
Run cycle group	10/24	10/24
Run cycle group with calendar offset (+ 3 days)	10/27/ (Saturday)	10/27/ (Saturday)
Run cycle group with free day rule	10/29/ (Monday)	0/29/ (Monday)
Run cycle in the job stream with time restrictions		
Run cycle in the job stream with + 4 working days shift	11/02 (Friday)	11/02 (Friday)
Run cycle in the job stream with free day rule	11/02 (Friday)	11/02 (Friday)
Run cycle in the job stream with earliest start +1 1pm	11/02 (Friday)	11/03 (Saturday) 1pm
Run cycle in the job stream without time restrictions		
Run cycle in the job stream with + 4 working days shift	11/02 (Friday)	11/02 (Friday) Start of Day

Table 11. Scenario 2. Time restriction in the run cycle group without offset (continued)

Run cycle group	Scheduled date	Earliest Start
Run cycle in the job stream with free day rule	11/02 (Friday)	11/02 (Friday) Start of Day

Table 12. Scenario 3. Time restriction in the run cycle group with offset (+1 12:00)

Run cycle group	Scheduled date	Earliest Start
Run cycle group	10/24	10/24
Run cycle group with calendar offset (+ 3 days)	10/27/ (Saturday)	10/27/ (Saturday)
Run cycle group with free day rule	10/29/ (Monday)	10/29/ (Monday)
Run cycle group with offset +1 12:00	10/29/ (Monday)	10/30 12:00 (Tuesday)
Run cycle in the job stream with time restrictions		
Run cycle in the job stream with + 4 working days shift	11/02 (Friday)	11/02 (Friday)
Run cycle in the job stream with free day rule	11/02 (Friday)	11/02 (Friday)
Run cycle in the job stream with earliest start +1 1pm	11/02 (Friday)	11/03 (Saturday) 1pm
Run cycle in the job stream without time restrictions		
Run cycle in the job stream with + 4 working days shift	11/02 (Friday)	11/03 12:00 (Saturday)
Run cycle in the job stream with free day rule	11/02 (Friday)	11/03 12:00 (Saturday)

z/OS Availability of the GENDAYS command at run cycle group level Using GENDAYS, you can check the result of the combination of all the run cycles in the group.

When you create a run cycle within a run cycle group, you must include it in a subset, identified by a subset ID.

Managing subsets of run cycles provides a lot of flexibility in job stream scheduling.

For example, you create run cycle group that contains three run cycles, two inclusive and one exclusive, in **OR** relationship:

RC1 Inclusive: run every Thursday of the month.

RC2 Inclusive: run every Friday of the month.

RC3 Exclusive: do not run the first week of each month.

In this case the result would be: the job stream runs every Thursday and Friday of the month except for those occurring in the first week of the month.

But if you want that the job stream also runs the first Thursday of the month, you could accomplish your goal by organizing your run cycles in subsets:

RC1 Inclusive: run every Thursday of the month.

RCSubset

RC2 Inclusive: run every Friday of the month.

RC3 Exclusive: do not run the first week of each month.

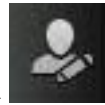
In this way the result would be: the job stream runs every Thursday of the month and every Friday of the month, except for the first Friday.

From the Workload Designer, you can run the following actions on run cycle groups:

- Create
- Create like
- Delete
- Edit
- View

To open the Workload Designer, do the following:

1. Log in to the Dashboard Application Services Hub.



2. From the navigation toolbar, click **Administration >Workload Design>Manage Workload Definitions**

In the displayed window, you are required to select an engine to connect with.

3. Enter the name of an engine and click **Go**.

The Workload Designer opens.

To create a run cycle group:

1. Click **New** in the Working List pane of the Workload designer window.
2. Select **Run Cycle group** in the resulting list of objects.

A Properties - Run Cycle group pane is opened on the bottom-right section of the window.
3. Enter the properties of the new run cycle group as requested. You can proceed to define the run cycles that are part of the group or leave this task for later.
4. To save your data, click the box next to the run cycle group name listed in the Working list pane and select the **Save selected** icon in the menu bar.

To exit the Properties - Run Cycle group pane without saving your data, click the box next to the run cycle group name listed in the Working list pane and select the **Delete selected** icon in the menu bar.

You can also create a run cycle group starting from an existing one. To do so, list the existing run cycle group and click the **Create Like** icon.

To run any of the other actions on already defined run cycle groups, you must first list the run cycle groups upon which you want to act. You can do this in one of the following ways:

- Click **Search** in the Working List pane:

1. Select **Run Cycle group** in the resulting list of objects.
The Search - Run Cycle group window is displayed.
 2. Click **Search** to display a list of all the run cycle groups defined on the engine. Before doing so, you can narrow your search by entering names or partial names in the text box.
The found run cycle group definitions are listed.
 3. Select the run cycle groups you want to act upon and click **Edit**, **View**, or **Delete**.
If you clicked **Edit** or **View**, the selected groups, inclusive of their run cycles, are listed in the Working List pane, with the properties of the top one displayed on the right pane.
If you clicked **Delete**, you are asked to confirm the deletion from a pop-up window.
- Click the **Run Cycle group** icon in the Quick Open pane:
 1. Click **Search** to display a list of all the run cycle groups defined on the engine. Before doing so, you can narrow your search by entering names or partial names in the text box.
The found run cycle group definitions are listed in the Quick Open pane.
 2. Select the run cycle groups you want to act upon and click **Edit** or **View**. You cannot delete from this pane.
The selected groups, inclusive of their run cycles, are listed in the Working List pane, with the properties of the top one displayed on the right pane.

To add run cycles to the group you are creating you must first include them in a subset, to do it perform the following steps:

1. From the Details view, right-click the run cycle group in the Details view and select **Add Subset**.
2. Right-click the subset and select **Add Run Cycle**.
3. Specify the run cycle attributes in the Run Cycle properties pane displayed at the bottom.

You can toggle between the panels displaying the properties of the run cycle group and of the included run cycles by selecting the entries listed in the **Type** column

To view, edit, or create another run cycle of a run cycle group, just click the corresponding icon in the Working List pane or in the Details pane.

When you create a group of run cycles, you can choose whether the run cycles must be in **AND** or in **OR** relationship. However, the result of the run cycles combined within the group must be a set of dates; the group cannot produce an exclusive result.

Related concepts:

"Run cycle" on page 41

"Run cycle group" on page 43

Related information:

"Run Cycle Preview" on page 221

Creating Workload Broker objects

You can manage Workload Broker jobs in a Tivoli Workload Scheduler environment by creating special workstations and job definitions.

Using the Dynamic Workload Console you can:

- Create the Workload Broker workstation in the Tivoli Workload Scheduler database.
- Create Workload Broker job definitions that map Workload Broker jobs.
- Monitor Workload Broker workstations in the plan.
- Submit jobs to the Workload Broker workstation.
- Monitor Workload Broker jobs and browse the corresponding job logs.

For more information about Tivoli Workload Broker, see *Tivoli Workload Scheduler Scheduling Workload Dynamically*.

To create a Workload Broker workstation, perform the following steps:



1. Click **Administration > Workload Environment Design > Create Workstations**.
2. Select a distributed engine from the list and click **Create Workstations**.
3. In the Workstations Properties panel, specify the following fields. The values you specify here must match the values specified during the installation of the Workload Broker component. You can retrieve them from the BrokerWorkstation.properties configuration file.

Name It is the value specified in the **TWS Agent name** field.

Workstation type

Select **Workload Broker** from the available workstation types.

Node Name

The host name of the computer where the Tivoli Workload Scheduler master is installed.

TCP/IP port

The port number you specified when installing the Tivoli Workload Scheduler master. It is the value specified in the **TWS Agent Port** field.

Leave all other fields with their default values.

4. Click **Save** to add the workstation to the database.

You can create Tivoli Workload Scheduler job definitions that map to Workload Broker jobs, and submit and monitor the job status. To create a new Workload Broker job definition, perform the following steps:



1. From the navigation toolbar, click **Administration > Workload Design > Manage Workload Definitions**.
2. In the Workload Designer, from the Working List pane click **New > Job Definition > Cloud > Workload Broker**.
3. In the Workspace pane, specify the properties for the job using the General, Task, Affinity, and Recovery Options tabs.
4. Click **Save** to add the Workload Broker job definition.

Related concepts:

“Workstation” on page 31

“Workload Broker job definition” on page 58


Creating an event rule

Use an event rule to specify a predefined set of actions in response to events that occur on the nodes where Tivoli Workload Scheduler runs. When one or more defined events occurs, you can specify which actions must be performed.

To create an event rule definition, perform the following steps:

1. Make sure you have the required authorizations granted as Dashboard Application Services Hub user, and in the Tivoli Workload Scheduler security file. For more information about security settings, see “Event management configuration” on page 9.



2. From the navigation toolbar, click **Administration > Workload Design > Create Event Rules**.
3. In the **Engine** field, specify the engine (this must be either the master domain manager or a node installed as backup master domain manager but not necessarily belonging to the master domain) where you want to run the event rule and click **Go**. Only after you have selected an engine connection, is the remainder of the panel displayed. If you have only one engine defined, it is displayed by default and the whole panel is displayed. Each time you create or modify an event rule, the corresponding object in the database is automatically locked. To release an object when you close the panel, click **Close** or the close button . You can unlock an event rule that you own by using the **Unlock** command. After closing the current rule, click **New** if you want to create a new rule without exiting the panel.

Note: When you change the engine connection, the available events and actions are also changed and the list of actions and events you had previously selected is reset. When you save your changes or select a different tab, a warning message is issued to inform you of the reset.

The Event Rule Editor panel displays the following main sections relating to:

- **General Information**
- **Events**
- **Actions**

Click the header bar of each section to minimize or expand it.

4. In the **General Information** section, enter the required information. All mandatory fields have a yellow background and are marked by an asterisk.

Note: Every time you modify and save any of these settings, the event rule (if not in **Draft** status) is newly deployed, and all possible events that were collected before the new deployment are ignored.

Rule name

Enter a name for the rule definition. It is a mandatory field with a maximum length of 40 characters. Blanks and special characters are not supported, except for minus (-) and underscore (_) characters. If you enter invalid characters, the field becomes red and the rule saving fails with an error.

Description

A generic descriptive text for the rule, with a maximum length of 120 characters.

Draft When an event rule is in **Draft** status, it is neither deployed nor

activated. When the event rule is not in **Draft** status, it is eligible to be deployed (automatically or manually, according to your deployment policy) and activated. As a result, actions are started whenever all the defined events take place, within the specified validity period. When you change a rule status into **Draft** or **Not Draft**, the rule goes respectively into Deactivation Pending or Activation Pending internal status.

Use the following options to specify time settings for the rule. The rule is active in the specified time zone, within the specified validity period during the days and hours defined.

Time zone

Select a time zone from the list. It is the time zone according to which the rule is active (regardless of the engine time zone). If no time zone is specified, the engine time zone is used by default.

Valid from...Valid to

Enter the time period during which the rule is valid. If the conditions specified in the rule are satisfied within this time period, the defined actions are run. You can also specify only one value, leaving the other field blank. Blank fields are considered as infinite.

Daily start/Daily end

It indicates the hours during which the rule is active, every day during the validity period specified. You can specify only one value, leaving the other field blank. The value of a blank field is considered midnight. You can also enter time values that cross midnight. For example, you can specify: a start time of 7 p.m. and an end time of 6 a.m. In this case the rule is active 24 hours a day, except from 7 p.m. to 6 a.m. everyday, for all the validity period.

5. In the **Events** section, select the events you want to monitor:
 - a. Each item of the list displayed on the left pane represents a category of events. Click an event category to view all the events belonging to it.
 - b. Click an event to automatically add it as a new box in the event section on the right pane. Within each of the boxes representing an event, click:



To open a new window listing the event properties and details.



To remove the event from the list.



The left or the right arrow at the bottom of the box to change the order in which the events occur. This is important when you define, within the event rule, a sequence of events.

- c. In the **Properties** section, specify the event properties.

The following list describes all the categories of events and their properties:

Note: PDF users, the parameter tables listed below are html files referenced by the PDF. They are not saved locally with the PDF from the infocenter. You must first view them on the infocenter before saving or printing.

Tivoli Workload Scheduler objects related events

All the events relating to scheduling objects such as jobs, job streams, workstations, and prompts. This type of event is described in more detail in Tivoli Workload Scheduler plan events.

Note: Any change performed on a workstation referenced in a rule is not reported in the rule. For example if you modify, update, or delete a workstation that is referenced in a rule, the rule ignores the change and continues to consider the workstation as it was when it was included in the rule.

File monitoring events

Events relating to changes to files and logs. File monitoring events are not supported on IBM i systems.

This type of event is described in more detail in File monitor.

Application monitoring events

Events relating to Tivoli Workload Scheduler processes, file system, and message box. Application monitoring events are not supported on IBM i systems.

This type of event is described in more detail in Application Monitor.

SAP related events

These events are available only if you have installed Tivoli Workload Scheduler for Applications and they are generated by external SAP systems. This type of event is described in more detail in SAP Monitor.

Generic events

Events used to manage custom events sent by external applications. You can write an XML file to define a custom event. A schema is provided to validate your XML, as well as a basic event template that you can use as a starting point. For more information, see the schemas for generic events. Events of this category are:

- Changes in a resource of the operating system, such as processes and memory
- Email received

- d. In the Events toolbar, choose an icon to specify the relationship between the events. The corresponding response actions are started only when the events take place according to this relationship. Click:



To specify one *single* event. Select this icon if you want a response action to start whenever the specified event occurs within the defined time period.



To specify a *set* of events, regardless of the sequence in which they occur. Select this icon if you want a response action to start if *all* the specified events occur, regardless of the order, within the defined time period.



To specify various events in *sequence*. Select this icon if you want a response action to start if *all* the specified events occur in sequence, within the defined time period.

- e. Optionally, in the event toolbar, you can specify a **Timeout period**. This is a further condition that can be set for the rule, only if you selected a *set* or *sequence* of two or more events. When you set the timeout option, the specified period is calculated starting from the first event that occurs; if all the remaining events occur *within* the specified timeout, the *response actions*

are started. If the timeout period expires before all the specified events have taken place, the defined *timeout actions* start.

- f. Optionally, in the **Correlate events on:** section, you can select some common properties.

These properties are common to the events selected. You can use them to create a correlation to further identify the events to monitor. The properties common to the selected events are automatically detected and listed in the right hand part of the event section. Select the properties you want to use for event correlation.

For example, if you want to receive an email every time a workstation remains unlinked for more than 10 minutes, you can correlate the following settings:

Event 1

Whatever workstation is unlinked.

Event 2

Whatever workstation is linked.

Event correlation

Start the action only if the property *workstation_name* is the same in Event 1 and Event 2.

Timeout

The two events must take place within 10 minutes of each other.

Action

Send an email to the administrator.

6. In the **Actions** section:

- a. Select the **Actions** tab. The number displayed in this tab, if any, represents the actions that have already been defined.
- b. In the left pane of this section, select the actions you want to start. Click an action to automatically add it to the action section on the right. Within each of the boxes representing an action click:



In the new panel that lists the action properties. To see an explanation about the action and its properties, hover with the cursor over the fields.



To remove the action from the list.

- c. In the **Properties** section, specify the action properties. The following list describes all the actions and their properties:

Operational actions

Actions that cause a change in the status of one or more Tivoli Workload Scheduler objects. Actions in this category include:

- Submitting jobs or job streams
- Submitting ad hoc jobs
- Replying to a prompt

This type of action is described in more detail in Tivoli Workload Scheduler actions.

- Adding an application occurrence (job stream) to the current plan on Tivoli Workload Scheduler for z/OS in Tivoli Workload Scheduler-Tivoli Workload Scheduler for z/OS end-to-end scheduling configurations.

This type of action is described in more detail in Tivoli Workload Scheduler for z/OS actions.

Notification actions

Actions such as:

- Sending emails or SMS. For details, see Mail sender plug-in
- Forwarding Tivoli Enterprise Console events. For details, see Message logger
- Writing messages in a logging repository. For details, see Tivoli Enterprise Console event forwarder
- Forwarding events to a Tivoli Business Services Manager server. For details, see Tivoli Business Services Manager Console event forwarder.
- Opening a ticket to a SmartCloud Control Desk. For details, see SmartCloud Control Desk.

Generic actions

Actions performed by running a command. This type of action is described in more detail in Generic action plug-in.

When defining the action properties, you can:

- Click **Variable** to associate the property to one or more variable values. At run time these values are displayed as properties associated to the selected action.
 - Only for Tivoli Workload Scheduler and Tivoli Workload Scheduler for z/OS Actions, you can click **Lookup** to select jobs, job streams, and workstations defined in the database.
- d. Optionally, select **Timeout actions** to define a further set of actions to start if the timeout period expires before all the specified events are performed. The number displayed in this tab, if any, represents the actions that have already been defined.

After you have defined events and actions, you can clear the **Draft** check box at the top of the panel and save the rule. The rule definition is then ready to be deployed and remains active according to its validity period, or until you disable it by marking it as **Draft**.

Related concepts:

“Event management” on page 64

“Event management configuration” on page 9

“Event rule” on page 231

Editing event rules

You can manage event rules in the database.

To modify an event rule definition in the database, perform the following steps.

1. From the navigation toolbar, click **Administration > Workload Design > Manage Event Rules**.
2. In the displayed panel, create a new task or run an existing task to retrieve a list of event rule definitions.
3. If required, specify the engine connections to run the task.

4. A table displays the list of event rule definitions available on the selected engine. Click a rule definition to view its properties or select it and click **Edit** to modify it. Alternatively, choose another action from the buttons and menu available.
5. In the displayed panel you can modify the current rule or create a new one.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.

6. Click **Save**.

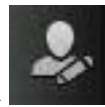
The event rule definition is now modified to the database.

Listing object definitions in the database

The following sections describe how you list the object definitions available in the database.

Listing workload objects

To list all the workload definitions except for event rules, open the Workload Designer as follows:



1. From the navigation toolbar, click **>Workload Design>Manage Workload Definitions**.
2. In the displayed panel, specify the engine connection you want to use. Only the categories of objects supported by the engine you selected are available.

You can open multiple occurrences of the Workload Designer, but only the following scenarios are supported:

- Same user connected to multiple engines
- Multiple users connected to the same engine.

From the Workload Designer you can use the following views to list objects:

Working List

Use this pane to search for objects in the database. Perform the following steps:

1. Open the **Search** menu and select the object you want to view or modify.
2. In the displayed lookup panel, select the required filtering criteria and click **Search**.
3. From the displayed list, select one or multiple objects and click one of the following action buttons:




Create like

To create a new object with the same properties of the selected object.



Edit

To modify the properties of the selected objects. When an object is open in edit mode the edit icon  is displayed on the right of the object.

**Unlock**

To unlock the selected objects for further actions. When an object is unlocked, it is displayed in read-only mode.

**Delete**

To delete the selected objects from the database.

New To create a new object.

Close To close the currently open object.

All the selected objects are listed in the Working List view. When you select an object from this list, its properties are displayed in the right pane, where you can view or edit them, if you have the required authorization defined in the Tivoli Workload Scheduler security file.

By hovering with the cursor over the icons located on the toolbar, you can see all the actions you can perform on the selected objects.

Quick Open

Use this pane to rapidly open objects from the database. Perform the following steps:

1. Click one of the icons displayed at the top of the pane to select the category of your search. Available categories depend on the engine connection you selected.
2. Refine your search by using some filtering criteria. You can filter by object name by entering the name, or part of it, in the text box, or by using the wildcard. Optionally, you can further filter your search by choosing more filtering criteria from the drop-down menu.
3. Click **Search**. The list of results is shown up to a maximum of 250 items. If the list is longer, use the filters to refine your search to reduce the number of results.
4. Select one or more objects and click **Edit**, to modify their properties, or **View**, to browse their properties.

You can drag the objects in the **Quick Open** pane and drop them in the **Graphical View** or you can quickly associate them to the item you are working with, by clicking the **Add** button. For example, you can search for jobs and automatically add them to the job stream you are editing, or add other objects as dependencies, such as resources or prompts.

Related concepts:

"Prompt" on page 50

"File" on page 50

"Resource" on page 50

"Calendar" on page 41

"User" on page 57

"Workstation class" on page 57

"Variable table" on page 58

Listing jobs and job streams

Distributed

To view a list of jobs and job streams available in the database, perform the following steps:



1. From the navigation toolbar, click **Administration > Workload Design > List Workload Definitions**.
2. In the displayed panel, specify your engine connection and the filtering criteria for your query. With distributed engine connections, from the **Object type** drop-down menu, select the object you want to list.
3. In the **Columns** section, select the column you want to display in the list of results.
4. Click **Display** to launch the query.

Related concepts:

"Job" on page 38

"Job stream" on page 38

Listing event rules

Distributed

To view a list of event rules available in the database, perform the following steps:



1. From the navigation toolbar, click **Administration > Workload Design > Manage Event Rules**. Or In the navigation bar on the left, click



System Status and Health > All Configured Tasks > New.

2. In the Select Task Type section, select **Event Management Task > List Event Rule Definitions** and click **Next**.
3. In the Enter Task Information panel, specify the task name and select the engine connection where you want to run the task. You can run this type of query only in a Tivoli Workload Scheduler distributed environment on either the master domain manager or on a node installed as a backup master domain manager, but not necessarily belonging to the master domain. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it.
4. Click **Next** to proceed with task creation or click **Finish** to complete the creation using the default values and exit without proceeding to the following steps. If you are editing an existing task, properties are organized in tabs.
5. In the Filter Criteria panel, specify some broad filtering criteria to limit the results retrieved by your query. Here you start refining the scope of your query by also considering the amount of information you want to retrieve. You can search for event rule definitions based on their status, validity intervals, or types of events and associated actions.
6. In the Columns Definition panel, select the information you want to display in the table containing the query results. According to the columns you choose here, the corresponding information is displayed in the task results table. For example, for each of the event rule definitions resulting from your query, you might want to see the status, type, or timeframe during which it is valid. You can then drill down into this information displayed in the table and navigate it. In the Columns Definition panel, not only can you select the columns for this task results, but you can also specify the columns for *secondary* queries on event rule instances. The information to be retrieved with these secondary queries is specified in this panel.

7. In the All Configured Tasks panel, you can see the main details about the task that you have just created. You can also choose to run the task immediately. In this case, if you have not yet specified an engine connection name, you must do so now. The task is now in the list of your tasks where you can open and modify it. You can find it in the task lists displayed by clicking the following options: **All Configured Tasks** or **Administration > Workload Design > Manage Event Rules**.
8. After running the task, a table is displayed showing a list of event rule definitions, on which you can take several actions, such as setting them as complete or as draft depending whether you want to make them available for deployment.

Listing workstations

To view a list of workstations available in the database, perform the following steps:

1. From the portfolio, click **Administration > Workload Environment Design > List Workstations**.
2. In the displayed panel, specify your engine connection and the filtering criteria for your query.
3. In the **Columns** section, select the column you want to display in the list of results.
4. Click **Display** to launch the query.

Creating a workload application template

To ensure the workload automation solution can be easily reproduced in another environment, there are some best practices to consider when creating the workload application template:

Job definitions

Jobs that refer to elements that are dependent on the environment or topology in which they reside, such as web service jobs, file transfer jobs, and database jobs to name a few, should make use of variables when specifying these elements such as credentials, paths, and port numbers. Variables can be managed in the mapping file so that the correct values can be assigned to the variable.

Workstation names

When jobs and job streams are extracted from the workload application during the export process, the names of the workstations are extracted as they are found in the source environment. Meaningful names or a standardized naming convention can simplify the mapping process.

Users Users are also extracted as they are found in the source environment. If the same user is not present in both source and target environment, then variables should be used to specify the user.

Mapping file

The mapping file should be maintained after performing the import process. It can be useful in the case where you want to replace a workload application or update it making the necessary changes to the mapping file.

Job stream variable table

All of the variables used to generically represent objects in the workload application should be added to a specific variable table related to the job stream in the workload application. This enables the customization of the job stream to reflect the target environment through the mapping file.

Avoid associating the default variable table to a job stream. The default variable table is extracted like any other table and will need to be renamed, otherwise, the import process fails because a table with the same name already exists. The target environment already has a default variable table, MAIN_TABLE, defined.

Run cycle variable table

All of the variables used to generically represent objects in the workload application should be added to a specific variable table related to the run cycle in the workload application. This enables the customization of the run cycle to reflect the target environment through the mapping file. Avoid associating the default variable table to a run cycle. The default variable table is extracted like any other table and will need to be renamed, otherwise, the import process fails because a table with the same name already exists. The target environment already has a default variable table, MAIN_TABLE, defined.

From the Workload Designer, you can create the template of a workload that can then be imported and run in another environment. You can create a workload application template containing one or more job streams with all the related jobs and internal or external dependencies (such as files, resources, prompts) so as to have a self-contained workflow. You can then export the workload application template to deploy and run it in another environment. To create a workload application template, perform the following procedure:



1. From the navigation toolbar, click **Administration > Workload Design > Manage Workload Definitions**
2. Specify the name of a distributed engine. The Workload Designer opens.
3. In the Working List pane, select **New > Workload Application Template**. The workload application template is created in the Details view and its properties page is displayed.
4. In the properties pane, specify the attributes for the workload application template that you are creating:

Name Mandatory field that contains the name of the workload application template. The maximum length is 80 characters.

Description

Optional descriptive text to help workload application users understand the purpose and characteristics of the workload application. The maximum length is 120 characters.

Provider

Optional field that specifies the creator of the workload application template. It can be useful to let workload application users know who created and provided it. The maximum length is 120 characters.

5. From the Details view, right-click the workload application template and click **Add Job Stream** to add job streams to it.
6. From the lookup dialog, select the job streams that you want to add. Together with the job streams, the corresponding dependencies are automatically also added to the workload application template.
7. Click **Save** to save the workload application template in the database.

8. Right-click the workload application template and click **Export** to produce a compressed file, named *workload application template name.zip*, containing all the files and information required to allow the workload to run also in another environment.

The compressed file contains:

*workload application template name*_Definitions.UTF8.xml

XML file that contains the definitions of all the exported objects. These definitions will be deployed in the target environment to populate the target database with the same objects existing in the source environment. The objects in the definition file can remain as they are or you can choose to rename them. If an object does not have a definition in the definition file, for example, a workstation, then at import time, a corresponding object will not be created in the target environment. The expectation is that such object is already present in the target environment, therefore, for these types of objects, you must map them in the mapping file.

*workload application template name*_Mapping.UTF8.properties

Mapping file that is used to replace the names of the objects in the source environment with the names that these objects have in the target environment. The objects that will be created in the target environment can be created with the same names as those in the source environment or you can specify a different name in this file.

*workload application template name*_SourceEnv_reference.txt

Reference information containing the definitions of the workstations used in the workload application template and other information that can be useful to correctly map the source environment into the target environment so as to allow the workload application to run.

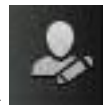
The compressed package must then be imported into the target environment where the workload application will be deployed, thus creating all the required objects in the target environment. In the target environment, the *workload application name*_Mapping.UTF8.properties file must be edited manually, using a generic text editor, by specifying the names of the objects as they are defined in the target environment (for example, the names of the workstations on which the job streams run). The import operation must be performed in the target environment by using the command line. For more details, see *Tivoli Workload Scheduler User's Guide and Reference* sections about workload applications and the wappman command.

Chapter 10. Changing user password in the plan

A User is the user name used as the login value for several operating system job definitions. Users are defined in the database and are associated to a password.

Users need to access the workstation where Tivoli Workload Scheduler launches jobs. If you need to change the user password after having already generated the plan, you can change the password in the plan. However, the changed password is only relevant to the current plan; Jnextplan restores the user password the next time it is run.

To change a user password in the plan, perform the following steps:



1. From the navigation toolbar, click **Administration > Workload Environment Design > Alter User Password in Plan**.
2. Select the engine.
3. Enter the following information in the Alter User Password in Plan panel:

Workstation

The name of the Tivoli Workload Scheduler workstation where the user can launch jobs.

User name

The user name. A user needs access to the workstation where Tivoli Workload Scheduler launches jobs, and have the right to **Log on as batch**.

The following formats are supported when specifying the value of the user name:

username

The Windows user. For example user1.

domain\username

The user belongs to a Windows domain. Specify the Windows domain name to which the user belongs. For example MYDOMAIN\user1.

username@internet_domain

The user belongs to an internet domain. The user name is in User Principal Name (UPN) format. UPN format is the name of a system user in an email address format. The user name is followed by the "at sign" followed by the name of the Internet domain with which the user is associated. For example administrator@bvt.com.

When a name is not unique it is considered to be a local user, a domain user, or a trusted domain user, in that order. If you schedule a job on a pool or a dynamic pool, the job runs with the user defined on the pool or dynamic pool. However, the user must exist on all workstations in the pool or dynamic pool where you plan to run the job.

Maximum length is 47 characters.

Password

The user password as defined on the computer. Maximum length is 31 characters.

Confirm password

The user password again for confirmation.

4. Modify the values as required and click **Save** to save the modified task.

Chapter 11. Monitoring your Objects in the Plan

To monitor your scheduling environment and workload objects in the plan, read the following sections.

Monitoring the progress of your plan

Request a graphical view of the progress of the current plan.

You can request a graphical view that shows the progress of the current plan on the engines for which you have configured a connection and specified the option to show the engine in the dashboard.

Tip: You can also request a graphical view of jobs and workstations from a mobile device. Refer to the *Mobile Applications User's Guide*.

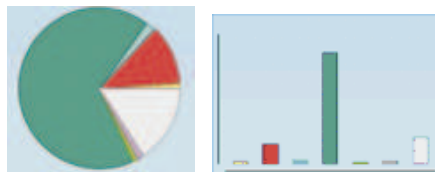


To request this graphical view, in the navigation bar on the left, click **System Status and Health > Workload Dashboard**. The panel opens showing a chart for each engine for which you configured a connection and specified the option to show the engine in the dashboard.



To customize the dashboard refresh time interval for all the engines, click **System Configuration > Workload Scheduling Settings > Set User Preferences**. In the lower part of this panel you can find a section to customize this interval.

Job status



The initial view is a pie chart, but optionally you can switch to the bar chart by clicking the icon in the toolbar below it. The pie is divided into slices of different colors that indicate, in the same way as the bars of the bar chart, the current status of jobs in the plan. Below the chart, a legend identifies the status that is mapped to each color and indicates the number of jobs in the plan that currently have that status. For more details about possible statuses and their meaning depending on the operating system, see the following sections in the *Dynamic Workload Console User's Guide*:

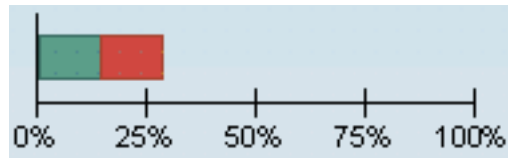
Distributed

Status description and mapping for distributed jobs

z/OS

Status description and mapping for z/OS job streams

Plan progress bar chart



The progress bar displayed in this section shows the progress of your plan. Successful jobs are represented by the green progress bar.



The progress bar view includes a subset of the job statuses included in the pie chart view. The following statuses are included:

- Successful
- Error

You can work with each pane in the following ways:

Change chart view

To switch between the pie chart and the bar chart view, click the graphic


icons in the toolbar below the graphic: , .

View details about the jobs that have a particular status


Click the hyperlinks to run a query to show details about the jobs with the selected status. The results of the query are displayed in a separate tab of Dynamic Workload Console. With engines version 8.6 and later, in case of rerun jobs, only the last rerun job instance is displayed in the query results.


View details about critical jobs


Critical jobs can be displayed only on engines on which the critical path

feature is available and enabled. Click the critical job icon  to view the section that displays the risk level of critical jobs on each engine. Click the hyperlinks to run a query on all the critical jobs with that risk level, running on that engine. The results of the query are displayed in a separate tab of Dynamic Workload Console

Critical jobs can have one of the following risk levels:

 **High Risk.** Critical jobs with high risk of missing their deadline. Their estimated end time is later than their deadline. This is probably because some predecessors have missed their deadline.

 **No Risk.** Critical jobs with no risk of missing their deadline. Their estimated end time is before their deadline. No action has to be taken.

 **Potential Risk.** Critical jobs with potential risk of missing their deadline. This is probably because some predecessors are late or are having a long duration process.

Related concepts:

Chapter 4, “Interact with your environment from a mobile device,” on page 13
Use your mobile device to easily and quickly interact with your Tivoli Workload Scheduler environment. You can monitor and run your workload from your mobile device by simply logging in to a URL and starting an app.

“Plans” on page 60

Monitor tasks

Create and run *monitor tasks* to obtain a list of objects on which you can perform monitoring and control actions. Creating and running a monitor task means creating a filter and running a search on it. This search returns a list of objects whose attributes satisfy the criteria that are defined in the task. From this list, you can click an item to view its properties or to run actions against it. In this way you can easily change some settings and the processing of the plan.

You can use the Dynamic Workload Console to create a monitor task, specifying your filter criteria by either selecting from a series of user interface options, or specifying a query in a query line.

If you have administrator's rights, you can configure a global setting so as to limit the number of results retrieved by monitor tasks. You can specify the maximum number of items that must be retrieved by the queries, and this configuration will apply to *all* monitor tasks run by current Dynamic Workload Console, except for Monitor Critical jobs. For more information, see “Limit the number of objects retrieved by queries” on page 190.

Related concepts:

“Monitoring your Scheduling Environment” on page 127

“Monitoring your Workload” on page 130

Creating a monitoring task query

Define a task to monitor objects in the plan by specifying a query in a query line. The objects for which you can create a monitoring task query are: jobs, job streams, workstations, files, resources, prompts, and domains.

To create a monitoring task query, perform the following steps.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.



1. In the navigation bar on the left, click **System Status and Health > Workload Monitoring > Direct Query**.

2. From the **Engine** drop-down list, specify the engine on which the task must run.

3. From the **Object Type** drop-down list, select the type of object you want to monitor. The object types available correspond to the type of engine selected: **Job, Job Stream, Workstation, File, Resource, Domain, Prompt**.

If you selected a z/OS engine, then you can create a monitoring task query for only the following objects: **Job, Job Stream, Workstation, Resources**.

4. In the **Query** text box, type a query containing the filter criteria to be applied to the object type selected. If you are familiar with the **conman** syntax, then the query syntax is quite similar, for example, the syntax for filtering jobs is similar to **conman showjobs**. If you are not familiar with the syntax, you can click Edit and specify the query by selecting the corresponding filter options from the user interface. In general:

- The plus symbol (+) is used to separate filter parameters.

- The hash key symbol (#) is used to separate the workstation name from the job stream name, for a distributed engine.
- The exclamation mark (!) is used to separate the workstation from the job stream name, for a z/OS engine.
- The ampersand symbol (@) replaces one or more alphanumeric characters (wild card).
- The period (.) separates the workstation name, job stream name from the job name.

For example, to display the status of all jobs in the acctg job stream on workstation site3 for a distributed engine, type the following string in the **Query** text box:

```
site3#acctg.@
```

or:

```
site3#acctg
```

See Example for more example queries and syntax. You can also select and edit an existing monitoring task from the **Query** drop-down list.

5. Click **Run** to run the task now. The results are displayed in table format in a new page. From this page you can continue to edit the query in the query line or click Edit to make changes to the user interface selections. The list of results are updated with the changes you make.
6. Alternatively, click **Edit** and select the filter criteria from a list of options rather than specify them in the **Query** text box. The **Query** text field is automatically populated with the corresponding query using the appropriate syntax.
7. Type a name for the monitoring task in the **Task Name** text box.
8. Click **Save** to save the task. Each task you create and save is included under the **All Configured Tasks** menu as well as in the **Query** drop-down list so they can be reused or modified as required.
9. Click **Run** to run the task. A list of results is displayed in table format in a new page.

You have created your task that, when run, creates a list of results satisfying your filtering criteria and showing, for each object in the list, the information contained in the columns you selected to view.

The following are some examples of the syntax used to create some specific queries in both a distributed and z/OS environment:

Job query in distributed environment

To query for all jobs, in all job streams and on all workstations on a specific distributed engine that have a dependency on a workstation with a name beginning with "FTA_1", and jobs beginning with "Job_A" in a job stream beginning with "JS_A", scheduled to run at 10 a.m. on October 31, 2013, that are in the "Waiting" state with a priority in the range of 1 - 50, specify the following query in the query line:

```
@#@.#+follows=FTA_1#JS_A@(1000 10/1/2013).JOB_A@+state=#Waiting,  
#Ready+priority=1,50
```

Job query in a z/OS environment


To query for all jobs in a job stream ending in "001", on all workstations beginning with "HR" in a z/OS environment, with an internal status of "Interrupted" and "Error" with a priority of "1", specify the following query in the query line:

Monitoring your Scheduling Environment

To monitor workstations and domains in your environment, you create and run *tasks*.

Note: You must create a connection to a remote Tivoli Workload Scheduler engine, before you can run tasks on it to obtain data.

When you create a task, you are actually defining a query where you specify multiple criteria to search for items and to display the search results. You can then save, reuse, and share this task with other users, and modify it at any time. When you run the task, you are actually running the query on the plan to retrieve the information according to the filters and level of detail you specified when you created the task.

Note: You can also save your tasks as one of your favorite bookmarks of your browser. To do this, from the panel displaying your task results, click the  icon to launch your task directly from the browser, just as you do when accessing any other website.

To create a task, perform the following procedure:



1. In the navigation bar on the left, click **System Status and Health > All Configured Tasks > New..**
2. In the Select Task Type panel, select the task you want to create, and click **Next**. You must select a task type to make the corresponding list active.
3. Follow the procedure relating to the specific task you are creating.

Each task you create and save is included under the **All Configured Tasks** menu.

Related concepts:

“Scheduling objects” on page 37

Creating a task to Monitor Workstations

To create a Monitor Workstations task, perform the following steps.

You can also define a task to monitor workstations by specifying and running a query based on the **conman showcpus** syntax from the Dynamic Workload Console. See “Creating a monitoring task query” on page 125.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.



1. In the navigation bar on the left, click **System Status and Health > All Configured Tasks > New..**
2. In the Select Task Type panel, under **Monitor Task**, select **Monitor Workstations** and click **Next**.

3. In the Enter Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.
4. Click **Next** to proceed with task creation or click **Finish** to complete the creation using the default values and exit without proceeding to the following steps. If you are editing an existing task, properties are organized in tabs.
5. In the General Filter panel, specify some broad filtering criteria to limit the results retrieved by your query. Here you start refining the scope of your query by also considering the amount of information you want to retrieve. Optionally, you can customize how often to refresh the information in the results table by specifying the refresh interval in seconds. If the periodic refresh is enabled for a task, when the task runs, the refresh time control options are shown in the results table. You can also set or change the periodic refresh interval directly in the results table. In this case, the value specified at task creation time is temporarily overwritten.

Distributed You can filter the task results based on the workstation and domain names, or part of names (using wildcard characters).

z/OS You can filter the task results based on the workstation types and reporting attributes.

6. In the Columns Definition panel, select the information you want to display in the table containing the query results. According to the columns you choose here, the corresponding information is displayed in the task results table. For example, for all the objects resulting from your query, you might want to see their link statuses, domains, and type, or you might want to see their statuses and the number of jobs successful or running on them. You can then drill down into this information displayed in the table and navigate it.

In the Columns Definition panel, not only can you select the columns for this task results, but you can also specify the columns for *secondary* queries on:

Distributed jobs, job streams domains, files, and resources. For example, you are creating a task to search for all the workstations of a domain. From the resulting list of workstations, you can navigate to see (secondary query) a list of all the jobs running on each of them.

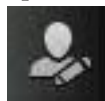
z/OS jobs. For example, you are creating a task to search for all the virtual workstations that are also fault-tolerant. From the resulting list of workstations, you can navigate to see (secondary query) a list of all the jobs running on each of them.

7. In the All Configured Tasks panel, you can see the main details about the task that you have just created. You can also choose to run the task immediately. In this case, if you have not yet specified an engine connection name, you must do so now. The task is now in the list of your tasks where you can open and modify it. You can find it in the task lists displayed by clicking the following

options:



System Status and Health > All Configured Tasks > or



Administration > Monitor > Monitor Workstations.

You have created your task that, when run, creates a list of workstations satisfying your filtering criteria and showing, for each workstation in the list, the information contained in the columns you selected to view.

Related concepts:

“Workstation” on page 31

Related reference:

“Workstation types” on page 206

Creating a task to Monitor Domains

Distributed

To create a Monitor Domains task, perform the following steps.

You can also define a task to monitor domains by specifying and running a query based on the **comman showdomain** syntax from the Dynamic Workload Console. See “Creating a monitoring task query” on page 125.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.



1. In the navigation bar on the left, click **System Status and Health > All Configured Tasks > New**.
2. In the Select Task Type panel, under **Monitor Task**, select **Monitor Domains** and click **Next**.
3. In the Enter Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.
4. Click **Next** to proceed with task creation or click **Finish** to complete the creation using the default values and exit without proceeding to the following steps. If you are editing an existing task, properties are organized in tabs.
5. In the General Filter panel, specify some broad filtering criteria to limit the results retrieved by your query. Here you start refining the scope of your query by also considering the amount of information you want to retrieve. Optionally, you can customize how often to refresh the information in the results table by specifying the refresh interval in seconds. If the periodic refresh is enabled for a task, when the task runs, the refresh time control options are shown in the results table. You can also set or change the periodic refresh interval directly in the results table. In this case, the value specified at task creation time is temporarily overwritten. You can filter the task results based on the domain name, or part of name (using wildcard characters). You can also configure the automatic refresh of the task results in the table.
6. In the Columns Definition panel, select the information you want to display in the table containing the query results. According to the columns you choose here, the corresponding information is displayed in the task results table. You can then drill down into this information displayed in the table and navigate it. In the Columns Definition panel, not only can you select the columns for this

task results, but you can also specify the columns for *secondary* queries on workstations. Starting from the Monitor Domains task table of result, you can start further queries on the workstations associated to one of the domains in the table; the information to be retrieved with these secondary queries is specified in this panel.

7. In the All Configured Tasks panel, you can see the main details about the task that you have just created. You can also choose to run the task immediately. In this case, if you have not yet specified an engine connection name, you must do so now. The task is now in the list of your tasks where you can open and modify it. You can find it in the task lists displayed by clicking the following



options: or **System Status and Health > All Configured Tasks > or**



Administration > Monitor > Monitor Domains.

You have created your task that, when run, creates a list of domains satisfying your filtering criteria and showing, for each domain in the list, the information contained in the columns you selected to view.

Related concepts:

“Domain” on page 34

The domain.

Monitoring your Workload

To control and manage scheduling activities and objects in plans, you create and run *tasks*.

Note: You must create a connection to a remote Tivoli Workload Scheduler engine, before you can run tasks on it to obtain data.

You can create the following types of task:

Monitor Task

When you create a Monitor task, you define a query where you specify multiple criteria to search for items and to display the search results. You can then save, reuse, and share this task with other users, and modify it at any time. Task sharing is enabled only if your role is authorized to share tasks. If you are not, contact the administrator or see: “Limit task and engine sharing” on page 191. When you run the task, you launch the query, according to the filtering criteria, on all the objects associated to the Tivoli Workload Scheduler connection you specified. A list of the objects that satisfy the search criteria is displayed when you run the task. You can view the objects resulting from your query, and their properties, and take actions on some of them.

See also “Monitoring your Scheduling Environment” on page 127 for information about other monitor tasks.

Event Management Task

When you create an Event Management Task you define a query where you specify multiple criteria to search for specific monitoring objects stored on the database and to display the search results. Available monitoring objects are event rules, triggered actions, and operator messages. You can then save, reuse, and share this task with other users, and modify it at any

time. When you run the task, you launch the query, according to the filtering criteria, on all the objects associated to the Tivoli Workload Scheduler connection you specified. A list of the objects that satisfy the search criteria is displayed when you run the task. You can view the objects resulting from your query, and their properties, and take actions on some of them.


Report Task

See Chapter 14, “Reporting,” on page 165 for information about this type of task.

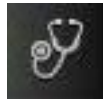
To create a task, perform the following procedure:



1. In the navigation bar on the left, click **System Status and Health > All Configured Tasks > New**.
2. In the Select Task Type panel, select the task you want to create, and click **Next**. You must select a task type to make the corresponding list active.
3. Follow the procedure relating to the specific task you are creating.

Note: You can also save your tasks as one of your favorite bookmarks in your browser. To do this, from the panel displaying your task results, click the add link icon  to launch your task directly from the browser, in the same way you do when accessing any other website.

Each task you create and save is included under the **All Configured Tasks** menu. Saved tasks are also listed under:



System Status and Health > Workload Monitoring

For tasks to monitor jobs, critical jobs, job streams, files, resources, prompts, and plan view.



System Status and Health > Event Monitoring > Workload Events

For tasks to monitor event rules, operator messages, and triggered actions.

Related concepts:

“Scheduling objects” on page 37

Event management tasks

You can create and run an event management task by creating a filter and running a search to get information about event management related objects.

The information retrieved when running event tasks is stored in the Tivoli Workload Scheduler databases, therefore, to run event tasks you must be connected to a Tivoli Workload Scheduler engine and must be authorized in the Tivoli Workload Scheduler security file to access those objects in the database.

You can create event management tasks to query for:

Event rule definitions

The template that defines an event rule consists of:

- One or more events, defined with its properties.

- The relationship between the specified events (they can be randomly grouped or listed in chronological order).
- The actions that must be performed when all the event conditions are satisfied.

Event rules

The instance of a rule definition in the plan.

Triggered action

The actual occurrence of an action defined in the event rule and triggered when the event conditions have been satisfied.

Operator messages

The instance of the **MessageLogger** action specified in the event rule definition. It provides information about the result of an event rule instance on a repository stored in the Tivoli Workload Scheduler relational database.

Related tasks:

“Creating a task to Monitor Event Rules” on page 145

“Creating a task to Monitor Triggered Actions” on page 147

“Creating a task to Monitor Operator Messages” on page 148

Plan view tasks

You can create and run *plan view tasks* to obtain a view of a collapsed plan showing its job streams, but hiding its jobs and dependencies. You can use this view to have an overall picture of the plan and how it is structured.

Creating and running a plan view task means creating a filter and running a search on it. This search returns a graphical representation of job streams whose attributes satisfy the criteria that are defined in the task.

From plan view, you can click a job stream to view its properties or to run some actions against it. You can also modify the filtering criteria to display different job streams.

Related tasks:

“Creating a task to Show Plan View” on page 157

Related information:

“Graphical Views in the plan” on page 210

Creating a task to Monitor Jobs

You can use this task to retrieve all jobs satisfying a number of filter criteria. For example, you can create a task to retrieve all jobs with a specific status, a specific priority, and by actual start or end time.

From the list of jobs displayed that satisfy your filter criteria, you can also retrieve a list of predecessors that are causing a delay of a selected job. Jobs in the predecessors list are in such states such as error, late, fence (for distributed systems only), suppressed (for distributed systems only) or long duration. If these jobs do not complete successfully on time, they prevent the selected job from completing on time. In this view, you can quickly see on which jobs you need to take appropriate recovery actions (for example, by releasing dependencies or answering prompts).

You can also define a task to monitor jobs by specifying and running a query based on the **command showjobs** syntax from the Dynamic Workload Console. See “Creating a monitoring task query” on page 125.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.

To create a Monitor Jobs task, perform the following steps.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.



1. In the navigation bar on the left, click **System Status and Health > All Configured Tasks > New**.
2. In the Create Task panel, under **Monitor Task**, select **Monitor Jobs** and click **Next**.
3. In the Enter Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.
4. Click **Next** to proceed with task creation or click **Finish** to complete the creation using the default values and exit without proceeding to the following steps. If you are editing an existing task, properties are organized in tabs.
5. In the General Filter panel, specify some broad filtering criteria to limit the results retrieved by your query. Here you start refining the scope of your query by also considering the amount of information you want to retrieve. Optionally, you can customize how often to refresh the information in the results table by specifying the refresh interval in seconds. If the periodic refresh is enabled for a task, when the task runs, the refresh time control options are shown in the results table. You can also set or change the periodic refresh interval directly in the results table. In this case, the value specified at task creation time is temporarily overwritten. You can search for jobs based on their status, on the workstation where they run, or the job streams they belong to. For example, you can look for all the jobs that did not complete successfully on the specified workstations, or you can look for all the jobs with the same owner and that have a specific priority level and an immediate restart option set.
6. In the Time Data Filter panel, specify a time range to limit your search to jobs or job streams that ran within a specific time period.

Note: If no date and time is specified then the jobs and job streams are not filtered based on their processing time.

7. **Distributed** In the Dependencies Filter panel, you can filter your results based on the dependencies they have. Only jobs that have the dependencies you specify in this panel are included in the query results. Dependencies can be jobs, job streams, resources, files, or prompts.
8. In the Columns Definition panel, select the information you want to display in the table containing the query results. According to the columns you choose

here, the corresponding information is displayed in the task results table. For example, for all the objects resulting from your query, you might want to see their statuses, the workstations where they ran, when they ran, and when they were scheduled to run. You can then drill down into this information displayed in the table and navigate it. In the Columns Definition panel, not only can you select the columns for this task results, but you can also specify the columns for *secondary* queries on job streams, jobs on critical network, and workstations. Starting from the Monitor Jobs task table of results, you can start further queries on the secondary objects associated to one of the jobs in the table; the information to be retrieved with these secondary queries is specified in this panel.

9. In the All Configured Tasks panel, you can see the main details about the task that you have just created. You can also choose to run the task immediately. In this case, if you have not yet specified an engine connection name, you must do so now. The task is now in the list of your tasks where you can open and modify it. You can find it in the task lists displayed by clicking the following



options: **System Status and Health > All Configured Tasks** or **Workload Monitoring > Monitor Jobs**.

You have created your task that, when run, creates a list of jobs satisfying your filtering criteria and showing, for each job in the list, the information contained in the columns you selected to view.

Related concepts:

“Job” on page 38

Related reference:

“Status description and mapping for distributed jobs” on page 199

“Status description and mapping for z/OS jobs” on page 201

“Limit the number of objects retrieved by queries” on page 190

Creating a task to Monitor Critical Jobs

You can use this task to retrieve all the jobs that were marked as **Critical** during their job stream creation. If it is critical that a job must be completed before a specific time, you can flag it as critical when you add it to a job stream using the Workload Designer. Jobs can also be flagged as critical by including the **critical** keyword to the job statement when you create or modify a job stream using the **composer** command line.

For more information about this, see the *Tivoli Workload Scheduler User's Guide and Reference*.

You can then use this list of critical jobs to control them, ensuring that nothing prevents them from completing on time.

Note: **Distributed** This task can only be run against the current plan and only on a master workstation.

Starting from the list of critical jobs, you can drill down and take actions on their predecessors (internal and external), which might be located faraway in the whole critical job network.

You can retrieve the following lists of predecessors to act on them (for example, by releasing dependencies or answering prompts) if they compromise the critical job success:

Critical Path

Critical job predecessors with the least slack time (delay allowed to let the critical job complete on time).

Hot List

The hot list contains a subset of critical predecessors that can cause a delay of the critical job because they are in such states as error, late, fence (for distributed systems only), suppressed (for distributed systems only) or long duration. If these jobs do not complete successfully on time, they prevent the critical job from completing on time. In the hot list view, you can quickly see which jobs need you to take appropriate recovery actions. Jobs included in the Hot List are not necessarily included in the Critical Path.

To create a Monitor Critical Jobs task, perform the following steps.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.



1. In the navigation bar on the left, click **System Status and Health > All Configured Tasks > New**.
2. In the Create Task panel, under **Monitor Task**, select **Monitor Critical Jobs** and click **Next**.
3. In the Enter Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.
4. Click **Next** to proceed with task creation or click **Finish** to complete the creation using the default values and exit without proceeding to the following steps. If you are editing an existing task, properties are organized in tabs.
5. In the General Filter panel, specify some broad filtering criteria to limit the results retrieved by your query. Here you start refining the scope of your query by also considering the amount of information you want to retrieve. Optionally, you can customize how often to refresh the information in the results table by specifying the refresh interval in seconds. If the periodic refresh is enabled for a task, when the task runs, the refresh time control options are shown in the results table. You can also set or change the periodic refresh interval directly in the results table. In this case, the value specified at task creation time is temporarily overwritten. You can search for jobs based on their status, on the workstation where they run, or on the job streams they belong to. For example, you can look for all the jobs that have a specific priority level and a high risk of missing their deadlines.

Note: The Monitor Critical Jobs task searches only for jobs that have been marked as *critical*.

Depending on what you choose as **Risk Level**, one or more of the following alert levels is shown in the list of critical jobs:



Critical jobs at high risk. This icon means that the critical job estimated end is beyond the job deadline. If nothing changes, the critical job is going to miss its deadline. The critical job estimated end is dynamically recalculated.



Critical jobs at potential risk. This icon means that the critical job estimated end has not yet passed the job deadline. However, the critical job has some predecessors in late, long duration, or error state. For distributed systems, the late condition can also be due to priority, limit, or fence values that are preventing jobs from running. If nothing changes, there is the possibility that the critical job will miss its deadline.



Critical job is on track. If nothing changes, it will meet its deadline.

6. In the Time Data Filter panel, specify a time range to limit your search to jobs or job streams that ran within a specific time period. If no date and time are specified, the jobs and job streams are not filtered based on their processing time.
7. In the Columns Definition panel, select the information you want to display in the table containing the query results. According to the columns you choose here, the corresponding information is displayed in the task results table. For example, for all the objects resulting from your query, you might want to see their statuses, the workstations where they ran, when they ran, and when they were scheduled to run. You can then drill down into this information displayed in the table and navigate it. In the Columns Definition panel, not only can you select the columns for this task results, but you can also specify the columns for *secondary* queries on jobs, job streams, jobs in critical network, and workstations. Starting from the Monitor Critical Jobs results, you can launch further queries on secondary objects associated to the jobs in the table. The information you can retrieve with these queries is specified in this panel. One of these secondary queries retrieves the list of **jobs in the critical network**, which includes *all* the critical job predecessors. The critical path is part of the critical network. The columns set for the list of **jobs in the critical network** are displayed as details of all the critical jobs Predecessors, and in the **Hot List** and **Critical Path** views. All these views can be launched by using the corresponding buttons from the Monitor Critical Jobs table of results.
8. In the All Configured Tasks panel, you can see the main details about the task that you have just created. You can also choose to run the task immediately. In this case, if you have not yet specified an engine connection name, you must do so now. The task is now in the list of your tasks where you can open and modify it. You can find it in the task lists displayed by clicking the following



options: **System Status and Health > All Configured Tasks** or **Workload Monitoring > Monitor Critical Jobs**.

You have created your task that, when run, creates a list of jobs satisfying your filtering criteria and showing, for each job in the list, the information contained in the columns you selected to view.

You can find a workload service assurance scenario in the *Tivoli Workload Scheduler User's Guide and Reference* about using this feature to monitor critical jobs.

Related concepts:

"Job" on page 38

"Workload service assurance" on page 68

Related reference:

"Status description and mapping for distributed jobs" on page 199

"Status description and mapping for z/OS jobs" on page 201

"Limit the number of objects retrieved by queries" on page 190

Related information:

"Using workload service assurance to monitor z/OS critical jobs" on page 173

Creating a task to Monitor Tasks on Multiple Engines

If you need to combine queries about jobs running on multiple distributed and z/OS engines, you can use Monitor Tasks on Multiple Engines. For example, it might be useful to run a query to find all the jobs in error in your environment by running a single task.

To create this task, perform the following steps.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.



1. In the navigation bar on the left, click **System Status and Health > All Configured Tasks > New**.
2. In the Create Task panel, select **Monitor Tasks on Multiple Engines > Monitor Jobs** and click **Next**.
3. In the Enter Task Information panel, specify a name for the task you are creating (a default name is provided) and define the Tivoli Workload Scheduler engines where you want to run the task. The task is run following the specified engine sequence. In this panel, you can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it.
4. Click **Next** to proceed with task creation or click **Finish** to complete the creation using the default values and exit without proceeding to the following steps. If you are editing an existing task, properties are organized in tabs.
5. In the General Filter panel, specify some broad filtering criteria to limit the results retrieved by your query. Here you start refining the scope of your query by also considering the amount of information you want to retrieve. Optionally, you can customize how often to refresh the information in the results table by specifying the refresh interval in seconds. If the periodic refresh is enabled for a task, when the task runs, the refresh time control options are shown in the results table. You can also set or change the periodic refresh interval directly in the results table. In this case, the value specified at task creation time is temporarily overwritten. You can search for jobs based on their identifier, on the job streams they belong to, or on their status, scheduled time, and priority. For example, you can look for all the jobs that did not complete successfully and were scheduled to run within a specific time period.
6. In the Columns Definition panel, select the information you want to display in the table containing the query results. According to the columns you choose


here, the corresponding information is displayed in the task results table. For example, for all the objects resulting from your query, you might want to see their statuses, the job streams they belong to, when they were scheduled to run, and the engines on which they ran. You can then drill down into this information displayed in the table and navigate it.

7. In the All Configured Tasks panel, you can see the main details about the task that you have just created. You can also choose to run the task immediately. In this case, if you have not yet specified an engine connection name, you must do so now. The task is now in the list of your tasks where you can open and modify it. You can find it in the task lists displayed by clicking the following



options: **System Status and Health > All Configured Tasks or Workload Monitoring > Monitor Jobs.**

You have created your task that, when run, creates a list of jobs satisfying your filtering criteria and showing, for each job in the list, the information contained in the columns you selected to view.

To view connection status information and statistical information about the engines against which the task was run, click the statistical icon  on the table toolbar.

A pie chart showing the number of query results and job status is displayed for each engine on which the task ran successfully. Click the pie sections to see further details. If the task did not run successfully on one or more engines, you see a message containing details about the errors.

Related concepts:

“Job” on page 38

“Monitoring jobs running on multiple engines” on page 175

Related reference:

“Status description and mapping for distributed jobs” on page 199

“Status description and mapping for z/OS jobs” on page 201

“Limit the number of objects retrieved by queries” on page 190

Creating a task to Monitor Job Streams

You can use this task to retrieve all job streams satisfying a number of filter criteria. For example, you can create a task to retrieve all job streams with a specific status, a specific priority, and by actual start or end time.

From the list of job streams displayed that satisfy your filter criteria, you can also retrieve a list of predecessors that are causing a delay of a selected job stream. Jobs in the predecessors list are in such states such as error, late, fence (for distributed systems only), suppressed (for distributed systems only) or long duration. If these jobs do not complete successfully on time, they prevent the selected job stream from completing on time. In this view, you can quickly see on which jobs you need to take appropriate recovery actions (for example, by releasing dependencies or answering prompts).

You can also define a task to monitor job streams by specifying and running a query based on the **conman showschedules** syntax from the Dynamic Workload Console. See “Creating a monitoring task query” on page 125.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.

To create a Monitor Job Streams task, perform the following steps.



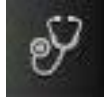
1. In the navigation bar on the left, click **System Status and Health > All Configured Tasks > New**.
2. In the Create Task panel, under **Monitor Task**, select **Monitor Job Streams** and click **Next**.
3. In the Enter Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.
4. Click **Next** to proceed with task creation or click **Finish** to complete the creation using the default values and exit without proceeding to the following steps. If you are editing an existing task, properties are organized in tabs.
5. In the General Filter panel, specify some broad filtering criteria to limit the results retrieved by your query. Here you start refining the scope of your query by also considering the amount of information you want to retrieve. Optionally, you can customize how often to refresh the information in the results table by specifying the refresh interval in seconds. If the periodic refresh is enabled for a task, when the task runs, the refresh time control options are shown in the results table. You can also set or change the periodic refresh interval directly in the results table. In this case, the value specified at task creation time is temporarily overwritten. You can search for job streams based on their scheduled run time, status, or the workstation where they run. For example, you can look for all the job streams with a given priority level that remained in waiting state on a set of workstations, or you can look for all the job streams with a specific group, a given priority level, and whose most critical job has a specified duration.
6. In the Time Data Filter panel, specify a time range to limit your search to jobs or job streams that ran within a specific time period.

Note: If no date and time is specified, then the jobs and job streams are not filtered based on their processing time.

7. **Distributed** In the Dependencies Filter panel, you can filter your results based on the dependencies they have. Only job streams that have the dependencies you specify in this panel are included in the query results. Dependencies can be jobs, job streams, resources, files, or prompts.
8. In the Columns Definition panel, select the information you want to display in the table containing the query results. According to the columns you choose here, the corresponding information is displayed in the task results table. For example, for all the objects resulting from your query, you might want to see their statuses, the workstations where they ran, when they ran, and when they were scheduled to run. You can then drill down into this information displayed in the table and navigate it. In the Columns Definition panel, not only can you select the columns for this task results, but you can also specify the columns for *secondary* queries on jobs and workstations (for distributed job streams only).

Starting from the Monitor Job Streams task table of result, you can start further queries on jobs and workstations associated to one of the job streams in the table; the information to be retrieved with these secondary queries is specified in this panel.

9. In the All Configured Tasks panel, you can see the main details about the task that you have just created. You can also choose to run the task immediately. In this case, if you have not yet specified an engine connection name, you must do so now. The task is now in the list of your tasks where you can open and modify it. You can find it in the task lists displayed by clicking the following



options: **System Status and Health > All Configured Tasks or Workload Monitoring > Monitor Job Streams.**

You have created your task that, when run, creates a list of job streams satisfying your filtering criteria and showing, for each job stream in the list, the information contained in the columns you selected to view.

Related concepts:

"Job stream" on page 38

Related reference:

"Status description and mapping for distributed job streams" on page 203

"Status description and mapping for z/OS job streams" on page 205

"Limit the number of objects retrieved by queries" on page 190

Creating a task to Monitor Job Streams on Multiple Engines

If you need to combine queries about job streams running on multiple distributed and z/OS engines, you can use Monitor Tasks on Multiple Engines. For example, it might be useful to run a query to find all the job streams running in your environment by running a single task.

To create this task, perform the following steps.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.



1. In the navigation bar on the left, click **System Status and Health > All Configured Tasks > New.**
2. In the Create Task panel, click **Monitor Tasks on Multiple Engines > Monitor Job Streams** and click **Next.**
3. In the Enter Task Information panel, specify a name for the task you are creating (a default name is provided) and define the Tivoli Workload Scheduler engines where you want to run the task. The task is run following the specified engine sequence. In this panel, you can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it.
4. Click **Next** to proceed with task creation or click **Finish** to complete the creation using the default values and exit without proceeding to the following steps. If you are editing an existing task, properties are organized in tabs.
5. In the General Filter panel, specify some broad filtering criteria to limit the results retrieved by your query. Here you start refining the scope of your query by also considering the amount of information you want to retrieve. Optionally,


you can customize how often to refresh the information in the results table by specifying the refresh interval in seconds. If the periodic refresh is enabled for a task, when the task runs, the refresh time control options are shown in the results table. You can also set or change the periodic refresh interval directly in the results table. In this case, the value specified at task creation time is temporarily overwritten. You can search for job streams based on their identifier, status, scheduled time, and priority. For example, you can look for all the job streams that are currently running and were scheduled to run within a specific time period.

6. In the Columns Definition panel, select the information you want to display in the table containing the query results. According to the columns you choose here, the corresponding information is displayed in the task results table. For example, for all the objects resulting from your query, you might want to see their statuses, when they were scheduled to run, and the engines on which they ran. You can then drill down into this information displayed in the table and navigate it.
7. In the All Configured Tasks panel, you can see the main details about the task that you have just created. You can also choose to run the task immediately. In this case, if you have not yet specified an engine connection name, you must do so now. The task is now in the list of your tasks where you can open and modify it. You can find it in the task lists displayed by clicking the following



options: **System Status and Health > All Configured Tasks or Workload Monitoring > Monitor Job Streams.**

You have created your task that, when run, creates a list of job streams satisfying your filtering criteria and showing, for each job stream in the list, the information contained in the columns you selected to view.

To view connection status information and statistical information about the engines against which the task was run, click the  statistical icon on the table toolbar.

A pie chart showing the number of query results and job stream status is displayed for each engine on which the task ran successfully. Click the pie sections to see further details. If the task did not run successfully on one or more engines, you see a message containing details about the errors.

Related concepts:

“Job stream” on page 38

Related reference:

“Status description and mapping for distributed job streams” on page 203

“Status description and mapping for z/OS job streams” on page 205

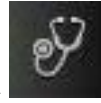
“Limit the number of objects retrieved by queries” on page 190

Creating a task to Monitor Files

To create a Monitor Files task, perform the following steps.

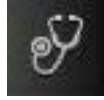
You can also define a task to monitor files by specifying and running a query based on the **conman showfiles** syntax from the Dynamic Workload Console. See “Creating a monitoring task query” on page 125.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.



1. In the navigation bar on the left, click **System Status and Health > All Configured Tasks > New**.
2. In the Create Task panel, under **Monitor Task**, select **Monitor Files** and click **Next**.
3. In the Enter Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.
4. Click **Next** to proceed with task creation or click **Finish** to complete the creation using the default values and exit without proceeding to the following steps. If you are editing an existing task, properties are organized in tabs.
5. In the General Filter panel, specify some broad filtering criteria to limit the results retrieved by your query. Here you start refining the scope of your query by also considering the amount of information you want to retrieve. Optionally, you can customize how often to refresh the information in the results table by specifying the refresh interval in seconds. If the periodic refresh is enabled for a task, when the task runs, the refresh time control options are shown in the results table. You can also set or change the periodic refresh interval directly in the results table. In this case, the value specified at task creation time is temporarily overwritten. You can filter the task results based on the file and workstation names, or part of names (using wildcard characters). You can also configure the automatic refresh of the task results in the table.
6. In the Columns Definition panel, select the information you want to display in the table containing the query results. According to the columns you choose here, the corresponding information is displayed in the task result. For example, for all the files resulting from your query, you might want to know their status and associated workstations. You can then drill down into this information displayed in the table and navigate it. In the Columns Definition panel, not only can you select the columns for this task results, but you can also specify the columns for *secondary* queries on jobs, job streams, and workstations. For example, you are creating a task to search for all the files whose names start with the same characters. From the resulting list of files, you can launch a secondary query to view a list of all the jobs associated to each of these files. Starting from the Monitor Files task results table, you can start further queries on secondary objects associated to one of the files in the table; the information to be retrieved with these secondary queries is specified in this panel.
7. In the All Configured Tasks panel, you can see the main details about the task that you have just created. You can also choose to run the task immediately. In this case, if you have not yet specified an engine connection name, you must do so now. The task is now in the list of your tasks where you can open and

modify it. You can find it in the task lists displayed by clicking the following



options: **System Status and Health > All Configured Tasks or Workload Monitoring > Monitor Files.**

You have created your task that, when run, creates a list of files satisfying your filtering criteria and showing, for each file in the list, the information contained in the columns you selected to view.

Related concepts:

“File” on page 50

Creating a task to Monitor Prompts

To create a Monitor Prompts task, perform the following steps.

You can also define a task to monitor prompts by specifying and running a query based on the **conman showprompts** syntax from the Dynamic Workload Console. See “Creating a monitoring task query” on page 125.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.



1. In the navigation bar on the left, click **System Status and Health > All Configured Tasks > New.**
2. In the Create Task panel, under **Monitor Task**, select **Monitor Prompts** and click **Next**.
3. In the Enter Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.
4. Click **Next** to proceed with task creation or click **Finish** to complete the creation using the default values and exit without proceeding to the following steps. If you are editing an existing task, properties are organized in tabs.
5. In the General Filter panel, specify some broad filtering criteria to limit the results retrieved by your query. Here you start refining the scope of your query by also considering the amount of information you want to retrieve. Optionally, you can customize how often to refresh the information in the results table by specifying the refresh interval in seconds. If the periodic refresh is enabled for a task, when the task runs, the refresh time control options are shown in the results table. You can also set or change the periodic refresh interval directly in the results table. In this case, the value specified at task creation time is temporarily overwritten. You can filter the task results based on the prompt name or status. You can also configure the automatic refresh of the task results in the table.
6. In the Columns Definition panel, select the information you want to display in the table containing the query results. According to the columns you choose here, the corresponding information is displayed in the task results table. For

example, for all the prompts resulting from your query, you might want to know their type and text. You can then drill down into this information displayed in the table and navigate it. In the Columns Definition panel, not only can you select the columns for this task results, but you can also specify the columns for *secondary* queries on jobs, and job streams. From the resulting list of prompts, you can launch a secondary query to view a list of all the jobs associated to each of them. Starting from the Monitor Prompts task results table, you can start further queries on secondary objects associated to one of the prompts in the table; the information to be retrieved with these secondary queries is specified in this panel.

7. In the All Configured Tasks panel, you can see the main details about the task that you have just created. You can also choose to run the task immediately. In this case, if you have not yet specified an engine connection name, you must do so now. The task is now in the list of your tasks where you can open and modify it. You can find it in the task lists displayed by clicking the following



options: **System Status and Health > All Configured Tasks or Workload Monitoring > Monitor Prompts.**

You have created your task that, when run, creates a list of prompts that satisfying your filtering criteria and showing, for each prompt in the list, the information contained in the columns you selected to view.

Related concepts:

“Prompt” on page 50

Creating a task to Monitor Resources

To create a Monitor Resources task, perform the following steps.

You can also define a task to monitor resources by specifying and running a query based on the **conman showresources** syntax from the Dynamic Workload Console. See “Creating a monitoring task query” on page 125.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.



1. In the navigation bar on the left, click **System Status and Health > All Configured Tasks > New.**
2. In the Create Task panel, under **Monitor Task**, select **Monitor Resources** and click **Next**.
3. In the Enter Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.
4. Click **Next** to proceed with task creation or click **Finish** to complete the creation using the default values and exit without proceeding to the following steps. If you are editing an existing task, properties are organized in tabs.

5. In the General Filter panel, specify some broad filtering criteria to limit the results retrieved by your query. Here you start refining the scope of your query by also considering the amount of information you want to retrieve. Optionally, you can customize how often to refresh the information in the results table by specifying the refresh interval in seconds. If the periodic refresh is enabled for a task, when the task runs, the refresh time control options are shown in the results table. You can also set or change the periodic refresh interval directly in the results table. In this case, the value specified at task creation time is temporarily overwritten. You can filter the task results based on the resource and workstation names, or part of names (using wildcard characters). You can also configure the automatic refresh of the task results in the table.
6. In the Columns Definition panel, select the information you want to display in the table containing the query results. According to the columns you choose here, the corresponding information is displayed in the task results table. For example, for all the resources resulting from your query, you might want to know if they are in use and the associated workstations. You can then drill down into this information displayed in the table and navigate it. In the Columns Definition panel, not only can you select the columns for this task results, but you can also specify the columns for *secondary* queries on jobs, job streams, and workstations. For example, you are creating a task to search for all the resources whose names start with the same characters. From the resulting list of resources, you can navigate to see (secondary query) a list of all the jobs that use each of them. Starting from the Monitor Task task results table, you can start further queries on secondary objects associated to one of the resources in the table; the information to be retrieved with these secondary queries is specified in this panel.
7. In the All Configured Tasks panel, you can see the main details about the task that you have just created. You can also choose to run the task immediately. In this case, if you have not yet specified an engine connection name, you must do so now. The task is now in the list of your tasks where you can open and modify it. You can find it in the task lists displayed by clicking the following



options: **System Status and Health > All Configured Tasks or Workload Monitoring > Monitor Resources.**

You have created your task that, when run, creates a list of resources satisfying your filtering criteria and showing, for each resource in the list, the information contained in the columns you selected to view.

Related concepts:

“Resource” on page 50

Creating a task to Monitor Event Rules

To create a Monitor Event Rules task, perform the following steps.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.



1. In the navigation bar on the left, click **System Status and Health > All Configured Tasks > New.**

2. In the Create Task panel, under **Event Management Task**, select **Monitor Event Rules** and click **Next**.
3. In the Enter Task Information panel, specify the task name and select the engine connection where you want to run the task. You can run this type of query only in a Tivoli Workload Scheduler distributed environment on either the master domain manager or on a node installed as a backup master domain manager, but not necessarily belonging to the master domain. Here you also specify whether to share the task with others, to allow them to see and run the task, but not to modify it.
4. Click **Next** to proceed with task creation or click **Finish** to complete the creation using the default values and exit without proceeding to the following steps. If you are editing an existing task, properties are organized in tabs.
5. In the General Filter panel, specify some broad filtering criteria to limit the results retrieved by your query. Here you start refining the scope of your query by also considering the amount of information you want to retrieve. Optionally, you can customize how often to refresh the information in the results table by specifying the refresh interval in seconds. If the periodic refresh is enabled for a task, when the task runs, the refresh time control options are shown in the results table. You can also set or change the periodic refresh interval directly in the results table. In this case, the value specified at task creation time is temporarily overwritten. You can search for rule instances based on their status, type, or trigger timestamps.
6. In the Columns Definition panel, select the information you want to display in the table containing the query results. According to the columns you choose here, the corresponding information is displayed in the task results table. For example, for all the objects resulting from your query, you might want to see their statuses, and the kind of rule that generated them. You can then drill down into this information displayed in the table and navigate it. In the Columns Definition panel, not only can you select the columns for this task results, but you can also specify the columns for *secondary* queries on event rules (the event rule definition stored in the database). The information to be retrieved with these secondary queries is specified in this panel.
7. In the All Configured Tasks panel, you can see the main details about the task that you have just created. You can also choose to run the task immediately. In this case, if you have not yet specified an engine connection name, you must do so now. The task is now in the list of your tasks where you can open and modify it. You can find it in the task lists displayed by clicking the following options: In the All Configured Tasks panel, you can see the main details about the task that you have just created. You can also choose to run the task immediately. In this case, if you have not yet specified an engine connection name, you must do so now. The task is now in the list of your tasks where you can open and modify it. You can find it in the task lists displayed by clicking



the following options: **System Status and Health > All Configured Tasks** or **Event Monitoring > Monitor Event Rules**.

You have created your query that, when run, lists the event rule instances satisfying your filtering criteria and shows, for each event rule in the list, the information you selected to view.

Related concepts:

“Event management tasks” on page 131

“Event management configuration” on page 9

Creating a task to Monitor Triggered Actions

To create a Monitor Triggered Actions task, perform the following steps.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.



1. In the navigation bar on the left, click **System Status and Health > All Configured Tasks > New**.
2. In the Create Task panel, select **Event Management Task > Monitor Triggered Actions** and click **Next**.
3. In the Enter Task Information panel, specify the task name and select the engine connection where you want to run the task. You can run this type of query only in a Tivoli Workload Scheduler distributed environment on either the master domain manager or on a node installed as a backup master domain manager, but not necessarily belonging to the master domain. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it.
4. Click **Next** to proceed with task creation or click **Finish** to complete the creation using the default values and exit without proceeding to the following steps. If you are editing an existing task, properties are organized in tabs.
5. In the General Filter panel, specify some broad filtering criteria to limit the results retrieved by your query. Here you start refining the scope of your query by also considering the amount of information you want to retrieve. Optionally, you can customize how often to refresh the information in the results table by specifying the refresh interval in seconds. If the periodic refresh is enabled for a task, when the task runs, the refresh time control options are shown in the results table. You can also set or change the periodic refresh interval directly in the results table. In this case, the value specified at task creation time is temporarily overwritten. You can search for triggered actions based on the type of rule instance that triggers them or on their scope. The scope of an action (or an event) is the set of properties that most characterize it.
6. In the Columns Definition panel, select the information you want to display in the table containing the query results. According to the columns you choose here, the corresponding information is displayed in the task results table. For example, for each of the actions resulting from your query, you might want to see the status, type, or associated message. You can then drill down into this information displayed in the table and navigate it. In the Columns Definition panel, not only can you select the columns for this task results, but you can also specify the columns for *secondary* queries on event rule instances. The information to be retrieved with these secondary queries is specified in this panel.
7. In the All Configured Tasks panel, you can see the main details about the task that you have just created. You can also choose to run the task immediately. In this case, if you have not yet specified an engine connection name, you must do so now. The task is now in the list of your tasks where you can open and modify it. You can find it in the task lists displayed by clicking the following



options: **System Status and Health > All Configured Tasks or Event Monitoring > Monitor Triggered Actions**.

You have created your query that, when run, lists the event rule instances satisfying your filtering criteria and shows, for each event rule in the list, the information you selected to view.

Related concepts:

“Event management tasks” on page 131

“Event management configuration” on page 9

Creating a task to Monitor Operator Messages

To create a Monitor Operator Messages task, perform the following steps.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.



1. In the navigation bar on the left, click **System Status and Health > All Configured Tasks > New**.
2. In the Create Task panel, select **Event Management Task > Monitor Operator Messages** and click **Next**.
3. In the Enter Task Information panel, specify the task name and select the engine connection where you want to run the task. You can run this type of query only in a Tivoli Workload Scheduler distributed environment on either the master domain manager or on a node installed as a backup master domain manager, but not necessarily belonging to the master domain. You also specify whether to share the task with others, to allow them to see and run the task, but not to modify it.
4. Click **Next** to proceed with task creation or click **Finish** to complete the creation using the default values and exit without proceeding to the following steps. If you are editing an existing task, properties are organized in tabs.
5. In the General Filter panel, specify some broad filtering criteria to limit the results retrieved by your query. Here you start refining the scope of your query by also considering the amount of information you want to retrieve. Optionally, you can customize how often to refresh the information in the results table by specifying the refresh interval in seconds. If the periodic refresh is enabled for a task, when the task runs, the refresh time control options are shown in the results table. You can also set or change the periodic refresh interval directly in the results table. In this case, the value specified at task creation time is temporarily overwritten. You can search for operator messages based on their severity, time stamp, or scope. The scope of an operator message is the set of properties that most characterize it.
6. In the Columns Definition panel, select the information you want to display in the table containing the query results. According to the columns you choose here, the corresponding information is displayed in the task results table. For example, for each of the operator messages resulting from your query, you might want to see the severity, the type of associated event, or the group in whose queue the message is. You can then drill down into this information displayed in the table and navigate it.
7. In the All Configured Tasks panel, you can see the main details about the task that you have just created. You can also choose to run the task immediately. In this case, if you have not yet specified an engine connection name, you must do so now. The task is now in the list of your tasks where you can open and modify it. You can find it in the task lists displayed by clicking the following



options: **System Status and Health > All Configured Tasks or Event Monitoring > Monitor Operator Messages.**

You have created your query that, when run, lists the event rule instances satisfying your filtering criteria and shows, for each event rule in the list, the information you selected to view.

Related concepts:

“Event management tasks” on page 131

“Event management configuration” on page 9

Controlling Jobs and Job Streams Processing

In the Dynamic Workload Console, you can control job and job streams processing by specifying dependencies and other properties as described in the following sections.

Using dependencies to control job and job stream processing

A *dependency* is a prerequisite that must be satisfied before processing can proceed. You can define dependencies for both jobs and job streams to ensure the correct order of processing. You can use these types of dependencies: **Distributed**

On completion of jobs and job streams

A job or a job stream must not begin processing until other jobs and job streams have completed. May be defined to require success or just completion

Resource

A job or a job stream needs one or more resources available before it can begin to run.

File

A job or a job stream needs to have one or more files meet the specified criteria before it can begin to run.

Prompt

A job or a job stream needs to wait for an affirmative response to a prompt before it can begin to run.

You can define up to 40 dependencies for a job or job stream. In a Tivoli Workload Scheduler network, dependencies can cross workstation and network boundaries.

z/OS

On completion of jobs belonging to the same job stream

A job must not begin processing until other jobs belonging to the same job stream have completed successfully.

On completion of jobs belonging to different job streams

A job must not begin processing until other jobs belonging to other job streams have completed successfully.

Resource

A job or a job stream needs one or more resources available before it can begin to run.

In addition to this, each job needs the workstation where it is scheduled to run to be available.

To add a dependency to a job or to a job stream from the Workload Designer, see “Adding and removing dependencies” on page 98.

You can also add a dependency from the panel displayed as result of your monitor task related to jobs or job streams by performing the following steps:

1. In the query result panel select a job or a job stream and click **Dependencies**.
2. In the Dependencies panel expand the section related to the dependency type you want to add and click **Add**.
3. Enter the required information and click **OK**.

For all the details about options and fields displayed in the panel, see the online help by clicking the question mark located at the top-right corner of the panel.

Using time restrictions to control job and job stream processing

Time restrictions can be specified for both jobs and job streams.

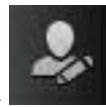
For a specific job or job stream you can specify the time that processing begins, **earliest start**, or the time after which processing can no longer be started, **latest start**. By specifying both, you define a time interval within which a job or job stream runs. You can use them as time dependencies.

You can also specify a **deadline** to specify the time within which a job or a job stream must complete. Jobs or job streams that have not yet started or that are still running when the deadline time is reached, are considered late in the plan. The deadline does not prevent jobs or job streams from starting.

You can also specify a **maximum duration** or a **minimum duration** for a job defined within a job stream. If a job is running and the maximum duration time has been exceeded, then the job can either be killed or can continue to run. If a job does not run long enough to reach the minimum duration time, then the job can be set to Abend status, to Confirm status awaiting user confirmation, or it can continue running.

For jobs you can also specify a **repeat range**; for example, you can have Tivoli Workload Scheduler launch the same job every 30 minutes between 8:30 a.m. and 1:30 p.m.

To specify time restrictions for a job or a job stream, perform the following steps:



1. From the navigation toolbar, click **Administration > Workload Design > Manage Workload Definitions**.
2. In the Workload Designer, edit the job stream you want to modify. For information about editing an object, see “Editing objects from the Working List View” on page 80.
3. Select the job or the job stream in the Details view and click the **Time restrictions** tab in the Properties section.
4. Enter the time restriction properties and save the job stream.

For all the details about options and fields displayed in the panel, see the online help by clicking the question mark located at the top-right corner of the panel.

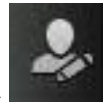
Using job priority and workstation fence to control distributed job processing

Distributed

Tivoli Workload Scheduler has its own queuing system, consisting of levels of *priority*. Assigning a priority to jobs gives you added control over their precedence and order of running.

The *fence* provides another type of control over job processing on a workstation. When it is set to a priority level, it only allows jobs whose priority exceeds the fence value to run on that workstation. Setting the fence to 40, for example, prevents jobs with priorities of 40 or less from being launched.

To specify job priority for a job, perform the following steps:



1. From the navigation toolbar, click **Administration > Workload Design > Manage Workload Definitions**.
2. In the Workload Designer, edit the job stream you want to modify. For information about editing an object see “Editing objects from the Working List View” on page 80.
3. Select the job in the Details view and click the **Scheduling options** tab in the Properties section.
4. Enter the job priority and save the job stream.
For all the details about options and fields displayed in the panel, see the online help by clicking the question mark located at the top-right corner of the panel.

You can also add a job priority from the panel displayed as result of your monitor task related to jobs by performing the following steps:

1. In the query result panel select a job and click **More Actions > Priority**.
2. In the Set Priority panel specify a priority value and click **OK**.

To set a workstation fence perform the following steps:

1. In the panel displayed as results of your monitor workstation task, select the workstation and click **More Actions > Fence**.
2. In the Set Fence panel specify a fence value and click **OK**.

Using limits to control job and job stream processing

The *limit* provides a means of setting the highest number of jobs that Tivoli Workload Scheduler is allowed to launch. You can set a limit:

- In the job stream definition
- In the workstation definition

Setting the limit on a workstation to 25, for example, allows Tivoli Workload Scheduler to have no more than 25 jobs running concurrently on that workstation.

To specify a limit for a job stream, perform the following steps:



1. From the navigation toolbar, click **Administration > Workload Design > Manage Workload Definitions**.
2. In the Workload Designer, edit the job stream you want to modify. For information about editing an object see “Editing objects from the Working List View” on page 80.
3. Select the job stream in the Details view and click the **Scheduling options** tab in the Properties section.
4. Enter the limit value and save the job stream.
For all the details about options and fields displayed in the panel, see the online help by clicking the question mark located at the top-right corner of the panel.

You can also add a limit from the panel displayed as result of your monitor job stream task by performing the following steps:

1. In the query result panel select a job stream and click **More Actions > Limit**.
2. In the Set Limit panel specify a new limit value and click **OK**.

To set a workstation limit perform the following steps:

1. In the panel displayed as result of your monitor workstation task, select the workstation and click **More Actions > Limit**.
2. In the Set Limit panel specify a new limit value and click **OK**.

Using job confirmation to control job processing

There might be scenarios where the completion status of a job cannot be determined until you have performed some tasks. You might want to check the results printed in a report, for example. In this case, you can set in the job definition that the job *requires confirmation*, and Tivoli Workload Scheduler waits for your response before marking the job as successful or failed.

To specify that a job requires confirmation, perform the following steps:



1. From the navigation toolbar, click **Administration > Workload Design > Manage Workload Definitions**.
2. In the Workload Designer, edit the job stream you want to modify. For information about editing an object see “Editing objects from the Working List View” on page 80.
3. Select the job in the Details view and click the **Scheduling options** tab in the Properties section.
4. Check **Requires confirmation** and save the job stream.
For all the details about options and fields displayed in the panel, see the online help by clicking the question mark located at the top-right corner of the panel.

Using job recovery actions to control job processing

When you schedule a job, you can specify the type of recovery you want performed by Tivoli Workload Scheduler if the job fails. The predefined recovery options are:

- Continue with the next job

- Stop and do not start the next job
- Run the failed job again

In addition, you can specify other actions to be taken in terms of recovery jobs and recovery prompts. For example, if a job fails, you can have Tivoli Workload Scheduler automatically run a recovery job, issue a recovery prompt that requires an affirmative response, and then run the failed job again.

To specify the job recovery actions, perform the following steps:



1. From the navigation toolbar, click **Administration > Workload Design > Manage Workload Definitions**.
2. In the Workload Designer, edit the job you want to modify. For information about editing an object see “Editing objects from the Working List View” on page 80.
3. Click the **Recovery options** tab in the Properties section.
4. Enter the recovery **Action** and the other information. Then save the job.
For all the details about options and fields displayed in the panel, see the online help by clicking the question mark located at the top-right corner of the panel.

Chapter 12. Working with Plans

This section contains the main tasks that involve plans. You can find information about selecting the working plan, creating trial and forecast plans, monitoring the progress of a plan, and generating a plan view.

Selecting the working plan

When you monitor the processing of your scheduling activities, you can choose the plan you want to work with. This plan is called the active plan.

There are several ways of defining the active plan. By default, the active plan is the default plan, which is associated to the engine connection. However, you can select a different plan and define it as active in the following ways:

“Associating a plan to the task”

Every time the task is run, it is run using the specified plan. Usually this setting is useful if you want to run multiple queries at the same time against different plans. To run the task on a different plan, you must either change this selection or use a temporary plan selection that lasts only until the current session expires. This plan selection takes precedence over all other selections and is the first to be used. It is made during the task creation when specifying the engine connection in the Enter Task Information panel.

“Dynamically defining a temporary plan” on page 156

You can set a temporary plan selection. This plan is used as the active plan for the current session only. All tasks and engines that have been set to use the active plan will use the currently defined plan. This plan selection is valid unless a different plan is defined as the active plan in the Enter Task Information panel.

“Associating a plan to an engine” on page 156

If none of the above settings are specified, tasks are run against the plan associated to the engine. By default, this plan and the active plan are the same.

Associating a plan to the task

To associate a plan to a task, perform the following steps:



1. In the navigation bar on the left, click **System Status and Health > All Configured Tasks > New**.
2. In the All Configured Tasks panel, select the type of task you want to create and click **Next**.
3. In the Enter Task Information panel, select a distributed engine as the scheduler engine where you want to run the task. If you select **Ask when needed**, you cannot perform the next steps.
4. Click **Select** to specify the plan that you want to associate to this task. Unless you change this selection, this task is always run against this plan.
Alternatively, click **Use Active Plan**, if you want this task to run using the plan that is set as the active plan in the plan list panel.

5. In the Select Plan panel, select the plan that you want to associate to this task and click **OK** to return to the Enter Task Information panel.

If you do not select any plan, the task uses the plan set as active. You can change the active plan by following the procedure described in “Dynamically defining a temporary plan.”

Dynamically defining a temporary plan

From the **Manage Available Plans** entry in the navigation toolbar, you can specify some filtering criteria to retrieve a list of plans.

Regardless of whether you are the owner of the engine connection or if it is only shared with you, you can select a plan from the list to define it as the active plan for the current session.

Follow these steps to define a filter and run a query to create a list of available plans:



1. From the navigation toolbar on the left, click **Planning > Workload Forecast > Manage Available Plans**.
2. In the Manage Available Plans panel:
 - a. Under **Select an Engine**, select the engine where you want to list the plans.
 - b. Under **Select Plan Type**, click the corresponding check box to select the type of plan you want to list. Selections are mutually exclusive. By default, all available plans are listed.
 - c. Under **Select Plan Name**, specify the name of the file containing the plan you want to search. You can use wildcard characters.
 - d. Click **Display Plans List** to generate a list of plans.
3. From the displayed list, select a plan and click **Set as active** to set it as a temporary plan that remains valid until the current session expires or you set another active plan.

Only for the current session, the plan selection made in this panel overrides the plan selection specified in the engine connection properties panel. However, it does not override the plan selection specified during the task definition.

Associating a plan to an engine

To associate a plan to an engine, perform the following steps:



1. From the navigation bar on the left, click **System Configuration > Manage Engines**.
2. In the Manage Engines panel, select an engine and click **Connection Properties**.
3. Under the Plans section, click **Select** to view a list of the available plans.
4. Choose a plan from the list and click **OK**. This setting is overridden by the plan selection made in the Enter Task Information panel or, if not available, by the selection made in the List Available Plans panel.

Related concepts:

“Plans” on page 60

Generating Trial and Forecast Plans

To create a new plan, perform the following steps:



1. From the navigation toolbar on the left, click **Planning > Workload Forecast**

Create Trial Plan

To create a trial plan. The Create Trial Plan panel is displayed.

Create Forecast Plan

To create a new forecast plan. The Create Forecast Plan panel is displayed.

2. Under the **Plan Information** section, enter the required information:

Engine Name

In the drop-down menu, select the engine where you want to create the plan. Only the engine connections that you created are available in the menu.

Plan Filename

Assign a name to the file that contains the plan. This is a mandatory field.

3. Under the **Plan Start** section, assign the date and time when the plan starts. Because the trial plan is mainly an extension of an existing and processing current plan, if you selected to create a new trial plan and a current plan is available on the engine, these fields are grayed out and their values are the same as the current plan end date. If you selected to create a new trial plan and a current plan is not available on the engine, or if you selected to create a new forecast plan, you can enter a date and time for the plan to start.
4. Under the **Plan End** section, assign one of the following values:
 - A date and time when the plan ends.
 - The number of days and hours the plan lasts.

By default the plan covers a one-day time interval.

5. Under the **Plan Time Zone** section, choose the time zone used in the plan.
6. Click **OK** to create the plan.

Related concepts:

"Plans" on page 60

Creating a task to Show Plan View

Note: This task can be run only on Tivoli Workload Scheduler v8.5.1 or later and Tivoli Workload Scheduler for z/OS v8.5.1 using the z/OS Connector v8.5.1 or later.

When you create a task, you define a query where you specify multiple criteria to search for items and to display the search results. You can then save, reuse, and share this task with other users, and modify it at any time. When you run the task, you run the query on the plan to retrieve the information according to the filters and level of detail you specified when you created the task.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.

To create a Show Plan View task, perform the following steps:



1. In the navigation bar on the left, click **System Status and Health > All Configured Tasks > New**.
2. In the Create Task panel, under **Monitor Task**, select **Show Plan View** and click **Next**.
3. In the Enter Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.
4. Click **Next** to proceed with task creation or click **Finish** to complete the creation using the default values and exit without proceeding to the following steps. If you are editing an existing task, properties are organized in tabs.
5. In the General Filter panel, specify some filtering criteria to limit the number of job streams displayed as a result. You can filter based on the names of job streams and workstation (distributed systems only), or on the starting times of the job streams. You can also decide whether to include predecessors and successors in the Plan View. Consider that predecessors and successors are included in the graphic only after all filter-matching job streams have already been included. When the maximum number of displayed job streams is reached, no more predecessors and successors are included in the graphic.
6. In the All Configured Tasks panel, you can see the main details about the task that you have just created. You can also choose to run the task immediately. In this case, if you have not yet specified an engine connection name, you must do so now. The task is now in the list of your tasks where you can open and modify it. You can find it in the task lists displayed by clicking the following



options: **System Status and Health > All Configured Tasks or Workload Monitoring > Show Plan View**.

You have created your task that, when run, generates the Plan View. The Plan View is a graphical representation of a filtered set of the job streams that make up the plan. Using the Plan View toolbar, you can perform several actions on the object displayed, as shown in the picture below: From the Plan View, you can also modify the filtering criteria to display different job streams, by clicking the view



and update filter icon. However, changes performed from this view apply to the current view only, they do not apply to the task saved in the task list.

Related concepts:

“Plans” on page 60

“Plan view tasks” on page 132

Related information:

“Graphical Views in the plan” on page 210

View preproduction plan

The preproduction plan is used to identify in advance the job stream instances and the job stream dependencies involved in a specified time period.

This improves performance when generating the production plan by preparing in advance a high-level schedule of the predicted production workload.

The preproduction plan contains:

- The job stream instances to be run during the time interval covered by the plan.
- The external dependencies that exist between the job streams and jobs included in different job streams.

From the Dynamic Workload Console you can view the preproduction plan graphically. You can open the preproduction plan in view mode only; you cannot extend it from this panel.

All users are allowed to view the preproduction plan. However, the preproduction plan content depends on the permissions you have on job streams. You can see only the job streams that you are allowed to see.

To open the preproduction plan view, perform the following procedure:



1. From the navigation toolbar on the left, click **Planning > View Preproduction Plan**.
2. In the displayed panel, select the distributed engine whose preproduction plan you want to view.
3. Optionally, specify a filter to reduce the number of job streams shown in the view. Only the job streams containing the string you entered as a filter are displayed in the preproduction plan view. By default, all job streams are shown. You can change your filtering criteria directly from the preproduction plan graphical view panel.
4. Specify the start and end dates to view only a portion of your preproduction plan. If you do not specify any date, the whole plan is shown. Optionally, you can organize the view by grouping the job streams by scheduled date.

The graphical view of the preproduction plan is displayed. By default, it shows a maximum number of 1000 job streams. If you want to change this setting, modify it in the "Override graphical view limits" on page 184 file. In this view you can view the job streams included in the plan together with their job stream dependencies. Each box represents a job stream, whose color indicates the status.

Right-click a job stream, to open the job stream definition within Workload Designer.

You can modify a job stream from the Workload Designer and then reload the updated preproduction plan view.

For more information about all the actions available from this view, refer to the online assistance, by clicking the "?" help button.

To open a job stream definition and view its details, right-click a job stream.

Related concepts:

Chapter 13. Submitting Workload on Request in Production

In addition to the jobs and job streams that are scheduled to run in production, you can also submit at any time other jobs and job streams. However, these jobs and job streams are not considered when identifying predecessors for jobs and job stream dependencies.

In production you can:

Distributed **Submit Ad Hoc Jobs**

This is a job that is:

- Not defined in the database.
- Used to run a command or a script in production.

Distributed **Submit Predefined Jobs**

This is a job that is:

- Defined in the database.

Submit Predefined Job Streams

This is a job stream that is:

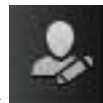
- Defined in the database.

See the following sections for instructions about inserting each of these types.

Submitting ad hoc jobs

Distributed

To add an ad hoc job to the current plan, perform the following steps:



1. From the navigation toolbar, click **Administration > Workload Submission > Submit Ad Hoc Jobs**.
2. In the displayed panel, select, from the drop-down list, the engine where you want to run the job, and click **Go**.
3. Enter all the required information about the job that you want to add. For more details about the information to enter in this panel, see the screen help, which you can open by clicking the question mark located at the top-right corner of the panel.
4. Click **OK** to save your changes, exit the panel and submit the job.

Submitting predefined jobs

Distributed

To add a predefined job to the current plan, perform the following steps:



1. From the navigation toolbar, click **Administration > Workload Submission > Submit Predefined Jobs**.

2. In the displayed panel, select, from the drop-down list, the engine where you want to run the job, and click **Go**.
3. Enter all the required information about the job that you want to add. For more details about the information to enter in this panel, see the screen help, which you can open by clicking the question mark located at the top-right corner of the panel.
4. Click **OK** to save your changes, exit the panel and submit the job.

Submitting predefined job streams

To add a predefined job stream to the current plan, perform the following steps:



1. From the navigation toolbar, click **Administration > Workload Submission > Submit Predefined Job Streams**.
2. In the displayed panel, select, from the drop-down list, the engine where you want to run the job.
3. Enter all the required information about the job stream that you want to add. For more details about the information to enter in this panel, see the screen help, which you can open by clicking the question mark located at the top-right corner of the panel. To find the job stream that you want to submit, you can launch searches based on part of the job stream name, workstation name, alias or variable table associated to it.
4. Optionally, specify a scheduled time to submit the job stream.
5. After completing the panel, click **Submit** to submit your job stream in the plan. Close the tab to exit discarding your changes.

Setting properties for ad hoc jobs and predefined jobs and job streams

To set the properties required to add jobs or job streams to the current plan, perform the following steps.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.

1. Enter the required information in the **General** section.
2. For predefined and ad hoc jobs only: under the **Task** tab, enter the task properties for the job in the displayed panel.
3. Select the **Time Restrictions** tab, and enter the required information in the displayed panel to set time restrictions for the job or job stream.
4. Select the **Resources** tab to set resource dependencies.
 - To create a new resource, click **New** and enter the required information in the Info panel.
 - To delete an existing resource, select it from the list and click **Delete**.
 - To modify a resource listed in the table, double-click its name and edit its properties in the Info panel.
5. Select the **Prompts** tab to set prompts as dependencies for the job or job stream.
 - To create a new prompt, click **New** and enter the required information in the Info panel.
 - To delete an existing prompt, select it from the list and click **Delete**.

- To modify a prompt listed in the table, double-click its name and edit its properties in the Info panel.
6. Select the **Files** tab to set file dependencies for the job or job stream.
 - To create a new file, click **New** and enter the required information in the Info panel.
 - To delete an existing file, select it from the list and click **Delete**.
 - To modify the file properties, double-click the file and edit the settings in the displayed table.
 7. Select the **Internetwork Predecessors** tab to add predecessor dependencies from a remote Tivoli Workload Scheduler network. The displayed panel shows existing internetwork predecessor properties.
 - To create a new internetwork predecessor, click **New** and enter the required information in the Info panel. Click the ... (Browse) button to search for and select the name of the network agent. Internetwork dependencies require that a network agent is configured to communicate with the external scheduler network.
 - To delete an existing internetwork predecessor, select it from the list and click **Delete**.
 - To modify an existing internetwork predecessor properties, double-click it, and edit its settings.
 8. Select the **Predecessors** tab to set predecessor dependencies for the job or job stream. The displayed panel shows existing predecessor properties.
 - To create a new predecessor, click **New** and enter the required information in the displayed panel.
 - To delete an existing predecessor, select it from the list and click **Delete**.
 - To modify an existing predecessor properties, double-click it, and edit its settings in the displayed table.

Chapter 14. Reporting

To create a report you must create and run *report tasks*.

Note: Before you can run a report to retrieve data from a Tivoli Workload Scheduler engine, you must complete the following steps:

1. Create a connection to a Tivoli Workload Scheduler engine.
2. Configure the Dynamic Workload Console to view report, as described in: Configuring Dynamic Workload Console to view reports

When you create a Report task, you generate Tivoli Workload Scheduler reports, which you can then view, print, and save, in different kinds of output. You can then save, reuse, and share this task with other users, and modify it at any time. You can also create customized reports by writing SQL scripts.

Related concepts:

“Reports” on page 66

Related reference:

“Reports” on page 224

Creating a task to generate a Job Run Statistics report

To create a task to run a Job Run Statistics Report, perform the following steps:



1. From the navigation toolbar, click **Workload Reporting > Create Historical Reports**, and click **New**.
2. In the Create Task panel, click **Report Definition > Job Run Statistics Report > Next**.
3. In the Enter Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.
4. In the Report Header panel, choose the name and the format of the output of the report, and click **Next** to proceed, or click **Finish** to complete the task creation using all the default values. If you are editing an existing task properties, click **Save** to keep your changes and open the next panel, if needed.
5. In the Filter Criteria panel, define a filter to select the jobs you want to include in the report. All the information about fields and options is available in the panel help. Click **Next** to proceed or click **Finish** to complete the task creation using all the default values.
6. In the Report Output Content panel, select the layout of your report. You can view the information either as a chart or as a table. The chart view displays the statistics for each job run in pie charts. You can select the report format. If you select HTML format, you can also limit the size of the report. You can also select the job details and the statistics you want to include in your report. After

you made your selection, click **Next** to proceed or click **Finish** to complete the task creation using all the default values.

7. In the Create New Task panel:
 - Select **Run this Task Now** and click **Finish** if you want to run the report task. If you have not yet specified an engine connection, you are prompted to do so now.
 - Click **Cancel** to exit without saving your changes.
 - Click **Finish** to save the task, without starting it immediately.

You have created your task that, when run, creates a report of jobs satisfying your filtering criteria and showing, for each job in the list, the information you selected to view.

Related concepts:

“Reports” on page 66

Creating a task to generate a Job Run History report

To create a task to run a Job Run History report, perform the following steps:



1. From the navigation toolbar, click **Workload Reporting > Create Historical Reports**, and click **New**.
2. In the Create Task panel, click **Report Definition > Job Run History Report > Next**.
3. In the Enter Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.
4. In the Report Header panel, choose the name and the format of the output of the report, and click **Next** to proceed, or click **Finish** to complete the task creation using all the default values. If you are editing an existing task properties, click **Save** to keep your changes and open the next panel, if needed.
5. In the Filter Criteria panel, define a filter to select the jobs you want to include in the report. All the information about fields and options is available in the panel help. Click **Next** to proceed or click **Finish** to complete the task creation using all the default values.
6. In the Report Output Content panel, select the layout of your report. You can view the information only as a table, but you can format it as an HTML or CSV file. If you select HTML format, you can also limit the size of the report. You can also select the job details you want to include in your report. After you make your selection, click **Next** to proceed or click **Finish** to complete the task creation using all the default values.
7. In the Create New Task panel:
 - Select **Run this Task Now** and click **Finish** if you want to run the report task. If you have not yet specified an engine connection, you are prompted to do so now.
 - Click **Cancel** to exit without saving your changes.
 - Click **Finish** to save the task, without starting it immediately.

You have created your task that, when run, creates a report of jobs satisfying your filtering criteria and showing, for each job in the list, the information you selected to view.

Related concepts:

“Reports” on page 66

Creating a task to generate a Workstation Workload Summary report

To create a task to run a Workstation Workload Summary report, perform the following steps:



1. From the navigation toolbar, click **Workload Reporting > Create Historical Reports**, and click **New**.
2. In the Create Task panel, select **Report Definition, Workstation Workload Summary Report**, and click **Next**.
3. In the Enter Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.
4. In the Report Header panel, choose the name and the format of the output of the report, and click **Next** to proceed, or click **Finish** to complete the task creation using all the default values. If you are editing an existing task properties, click **Save** to keep your changes and open the next panel, if needed.
5. In the Filter Criteria panel, define a filter to select the jobs you want to include in the report. All the information about fields and options is available in the panel help. Click **Next** to proceed or click **Finish** to complete the task creation using all the default values.
6. In the Report Output Content panel, select the layout of your report. You can view the information either as a chart or as a table. The chart view shows the workload of all the specified workstations, aggregated by time. You can also choose to view the workloads of all the specified workstations in a single line chart. Aggregating all the information, you have a comparative view of the workstation workloads. You can select the report format. If you select HTML format, you can also limit the size of the report. You can also select the granularity with which to extract the data (by day or hour) and to order the report. After you make your selection, click **Next** to proceed or **Finish** to complete the task creation using all the default values.
7. In the Create New Task panel:
 - Select **Run this Task Now** and click **Finish** if you want to run the report task. If you have not yet specified an engine connection, you are prompted to do so now.
 - Click **Cancel** to exit without saving your changes.
 - Click **Finish** to save the task, without starting it immediately.

You have created your task that, when run, creates a report of jobs satisfying your filtering criteria and showing, for each job in the list, the information you selected to view.

Related concepts:

Creating a task to generate a Workstation Workload Runtimes report

To create a task to run a Workstation Workload Runtimes report, perform the following steps:



1. From the navigation toolbar, click **Workload Reporting > Create Historical Reports**, and click **New**.
2. In the Create Task panel, select **Report Definition, Workstation Workload Runtimes Report**, and click **Next**.
3. In the Enter Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.
4. In the Report Header panel, choose the name and the format of the output of the report, and click **Next** to proceed, or click **Finish** to complete the task creation using all the default values. If you are editing an existing task properties, click **Save** to keep your changes and open the next panel, if needed.
5. In the Filter Criteria panel, define a filter to select the jobs you want to include in the report. All the information about fields and options is available in the panel help. Click **Next** to proceed or click **Finish** to complete the task creation using all the default values.
6. In the Report Output Content panel, select the layout of your report. You can view the information either as a chart or as a table. The chart view shows in a bar chart the number of jobs that are running on the workstations. Selecting the chart view, you can also specify how many jobs to display in each chart. You can specify the format of your report. If you select HTML format, you can also limit the size of the report. You can also select the information you want to include in the report and how you want it ordered. After you make your selection, click **Next** to proceed or **Finish** to complete the task creation using all the default values.
7. In the Create New Task panel:
 - Select **Run this Task Now** and click **Finish** if you want to run the report task. If you have not yet specified an engine connection, you are prompted to do so now.
 - Click **Cancel** to exit without saving your changes.
 - Click **Finish** to save the task, without starting it immediately.

You have created your task that, when run, creates a report of jobs satisfying your filtering criteria and showing, for each job in the list, the information you selected to view.

Related concepts:

“Reports” on page 66

Creating a task to Create Plan Reports

Perform the following steps to create one of the following reports:

Actual Production Details Report

A report based on the information stored either in the current or in an archived plan. The information contained in these plans is retrieved from the Symphony files. Actual Production Details Report can be run on distributed engines (master domain manager, backup domain manager, domain manager with connector, and fault-tolerant agent with connector).

Planned Production Details Report

A report based on the information stored either in a trial or in a forecast plan. The information contained in these plans is retrieved from the Tivoli Workload Scheduler database. A Planned Production Details Report can be run on distributed engines (master domain manager and backup domain manager). A real production report extracted from a fault-tolerant agent might contain different information with respect to a plan extracted from a master domain manager. For example, the number of jobs and job streams is the same, but their status can change, because a job successful on the master can be in hold or ready on the agent. The update status rate is the same only on the full status agent that runs on the domain master.



1. From the navigation toolbar, click **Workload Reporting > Create Plan Reports**, and click **New**.
2. In the Create Task panel, choose the type of report that you want to create, and click **Next**.
3. In the Enter Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.
4. In the Report Header panel, choose the name and the format of the output of the report and click **Next** to proceed or click **Finish** to complete the task creation using all the default values.
5. In the Filter Criteria panel, define a filter to select the jobs that you want to include in the report. All the information about fields and options is available in the panel help. Click **Next** to proceed or click **Finish** to complete the task creation using all the default values.
6. In the Report Output Content panel, select the job information that you want to display in the report output. After you make your selection, click **Next** to proceed or **Finish** to complete the task creation using all the default values.
7. In the Create New Task panel:
 - Select **Run this Task Now** and click **Finish** if you want to run the report task. If you have not yet specified an engine connection, you are prompted to do so now.
 - Click **Cancel** to exit without saving your changes.
 - Click **Finish** to save the task, without starting it immediately.

You have created your task that, when run, creates a report of job and job streams processing details satisfying your filtering criteria.

Related concepts:

“Reports” on page 66

Creating a task to Create Custom SQL Reports

Use this task to define your own reports by writing or importing SQL queries for extracting data in HTML or CSV format. To create an SQL report task, perform the following steps:



1. From the navigation toolbar, click **Workload Reporting > Create Custom SQL Reports**, and click **New**.
2. In the Create Task panel, select **Report Definition > Custom SQL Report Task > Next**.
3. In the Enter Task Information panel, define the type of scheduler engine here you want to run the task. You can select an engine at a later time. Remember that the engine name must be specified before running the task. Depending on the engine type you choose, the filtering criteria and the results you can display are different. You can also specify whether to share the task with others, to allow them to see and run the task, but not to modify it. Task and engine sharing can be disabled by the TWSWEBUIAdministrator in the global settings customizable file.
4. In the Report Header panel, choose the name and the format of the output of the report and click **Next** to proceed or **Finish** to complete the task creation using all the default values. The **Custom SQL Report** supports only the **Tables View** in either HTML or CSV format.
5. In the Filter Criteria panel, enter the SQL statement on which you want to base your report. You can write the query in the text pane or load an existing query from a file by browsing to the required file and clicking **Load**. Click **Next** to proceed or **Finish** to complete the task creation using all the default values.
6. In the Report Output Content panel, select the job information that you want to display in the report output. After you make your selection click **Next** to proceed or **Finish** to complete the task creation using all the default values.
7. In the Create New Task panel:
 - Select **Run this Task Now** and click **Finish** if you want to run the report task. If you have not yet specified an engine connection, you are prompted to do so now.
 - Click **Cancel** to exit without saving your changes.
 - Click **Finish** to save the task, without starting it immediately.

You have created your task that, when run, creates a report satisfying your filtering criteria and showing the information you selected to view.

Related concepts:

"Reports" on page 66

Related information:

"SQL report examples" on page 229

Chapter 15. Scenarios

This section provides some scenarios about the product usage that can help you get familiar with Tivoli Workload Scheduler.

You can find additional scenarios about the product usage at the following links:

- Tivoli Workload Scheduler Wiki Media Gallery, to access demos (only available in English) about how to use the Tivoli Workload Scheduler product
- *Scenarios and How to Demos*, to access scenarios about dynamic scheduling and variable table usage
- A workload service assurance scenario in the *Tivoli Workload Scheduler User's Guide and Reference* about using this feature to monitor critical jobs.

Customizing your job stream

This scenario describes how you can use the Dynamic Workload Console to create a job stream, schedule it to run at specified times, and change its behavior based on the day when it is scheduled to run.

Overview

A sales department manager needs to collect data on sales reports both at business unit level and at corporation level. For this reason the department manager needs reports both on a weekly and on a monthly basis. Report data is stored in two different directories, as follows:

- Data for weekly reports is stored in a set of files located in the directory `/reports/weekly`.
- Data for monthly reports is stored in a set of files located in the directory `/reports2/monthly`.

The job stream used for generating the reports has a dependency on the presence of these files. To collect the required data, the Tivoli Workload Scheduler administrator creates one job stream with two different run cycles, one scheduled to run on a weekly basis and one scheduled to run on a monthly basis.

Each run cycle references two different variable tables, which contain the variable and the related value used to define the path where the correct input files are located.

Creating the job stream and the related objects

To create all the database objects required to reach the business goal, the Tivoli Workload Scheduler designer uses the Workload Designer on the Dynamic Workload Console.

1. He logs in to the Dynamic Workload Console and, from the navigation toolbar



on the left, he clicks **Administration > Workload Design > Manage Workload Definitions**. The Workload Designer opens.

Using the **New** menu in the **Working List** section, the administrator can create all the necessary objects. He can also search for existing objects to edit and insert in the plan in the **Quick Open** section.

2. The administrator selects **New > Variable Table** to create the two variable tables required to provide the two different values for the input file paths.
 - a. He creates a variable table with name SC1_WEEKLY_DATA_TABLE. This table is the default table. The path to the files required to generate the weekly reports is indicated by the REP_PATH variable, to which he assigns the value "/reports/weekly".
 - b. He creates a variable table with name SC1_MONTHLY_DATA_TABLE. The path to the files required to generate the monthly reports is indicated by the REP_PATH variable, to which he assigns the value "/reports2/monthly".
3. The administrator selects **New > Job Definition > Windows Job Definition** to create the job definitions which generate the reports. All job definitions run a script each, which receives the value of the REP_PATH variable as the input value. He creates the following job definitions:
 - a. The job definition named SC1_PARSE_DATA SCRIPTNAME runs on the relevant workstation logging in as root. It runs a script which contains the following statement: "/reportApp/parseData.sh ^REP_PATH^".
 - b. The job definition named SC1_PROCESS_DATA SCRIPTNAME runs on the relevant workstation logging in as root. It runs a script which contains the following statement: "/reportApp/processData.sh ^REP_PATH^".
 - c. The job definition named SC1_CREATE_REPORTS SCRIPTNAME runs on the relevant workstation logging in as root. It runs a script which contains the following statement: "/reportApp/createReports.sh ^REP_PATH^".
4. The administrator selects **New > Job Stream** to create the job stream which contains the jobs. The job stream is named SC1_RUN_REPORTS and runs on the relevant workstation.
5. The administrator selects **Add to Selected > Run Cycle > Inclusive** to define two run cycles for the job stream, as follows:
 - a. The run cycle named SC1_WEEKLY_RCY uses the variable table SC1_WEEKLY_DATA_TABLE, which contains the value for the file path to be used for generating the weekly report. The run cycle also specifies that the job stream is to run once per week on Friday.
 - b. The run cycle named SC1_MONTHLY_RCY uses the variable table SC1_MONTHLY_DATA_TABLE, which contains the value for the file path to be used for generating the monthly report. The run cycle also specifies that the job stream is to run once per month on the 27th day.
6. The administrator selects **Add to Selected > Add Dependency > File** to specify a dependency from the files containing the data used for report generation. He uses the REP_PATH variable to define the path to the required files.
7. The administrator searches for the job definitions he previously created (SC1_PARSE_DATA SCRIPTNAME, SC1_PROCESS_DATA SCRIPTNAME, SC1_CREATE_REPORTS SCRIPTNAME) and adds them to the job stream.
8. The administrator creates a plan lasting 30 days to generate multiple instances of the job stream.

As a result, the variable REP_PATH is assigned different values depending on the run cycle that applies. The administrator defines two run cycles, each of which references a specific variable table.

In this way, the job stream instances have a dependency on a different set of files depending on the type of report they have to produce, either monthly or weekly, as follows:

- The job stream instances that generate the weekly report have a dependency on the files located in the /reports/weekly directory.
- The job stream instances that generate the monthly report have a dependency on the files located in the /reports2/monthly directory.

Furthermore, the name of the target directory is correctly replaced in the task string of the three jobs run by every job stream instance, as follows:

- The jobs run by job stream instances that generate the weekly report run shell scripts with the directory /reports/weekly as an input argument.
- The jobs run by job stream instances that generate the monthly report run shell scripts with the directory /reports2/monthly as an input argument.

The administrator can therefore define a single job stream with two different run cycles and ensure that the appropriate reports are generated on the correct dates without further user intervention.

Related concepts:

“Job stream” on page 38

“Job” on page 38

“Variable table” on page 58

“Run cycle” on page 41

“Dependencies” on page 48

Controlling processing using dependencies

Related tasks:

“Designing your Workload” on page 79

“Creating job definitions” on page 84

“Prerequisite steps to create job types with advanced options” on page 91

Using workload service assurance to monitor z/OS critical jobs

z/OS

This scenario shows how an operator can monitor the jobs that are critical for the customer business and that must complete by their deadline.

Overview

The operator uses the Dynamic Workload Console to meet a Service Level Agreement (SLA) that requires a DB2 database to be up and running each day by 3 p.m., after its backup.

The operator must be informed whether critical jobs risk missing their deadline, to take the appropriate actions if needed. While the plan is running, the operator expects that the scheduler dynamically controls the network of submitted jobs, detecting when a critical job predecessor is late, long running, or ended with an error.

Roles

The scheduling administrator and the operator are involved in this scenario:

Tivoli Workload Scheduler for z/OS scheduling administrator

When planning the operations, the administrator defines:

- Scheduled time, duration, and deadline times.
- Critical jobs.

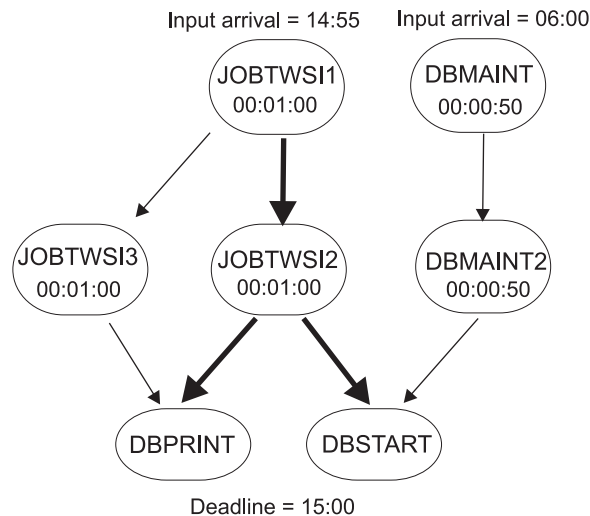
Tivoli Workload Scheduler operator

Controls the submitted workload by using Critical Jobs and Hot List views.

Setting up the environment

Complete the following tasks when planning your operations:

1. Mark your critical jobs in the z/OS database. Set DBSTART and DBPRINT as critical jobs, using a job network with the following structure:



Key:

- Job dependency
- Job dependency on a critical path

2. Run a daily planning job. The daily planning process calculates the critical paths in your job network, using the deadline, scheduled time arrival, and duration settings.

Running the scenario

After you updated your current plan, you can monitor your critical workload by using **Critical Path** and **Hot List** views:



1. In the navigation bar on the left, click **System Status and Health > Workload Monitoring > Monitor Critical Jobs**.
2. Click **New** to create a Monitor Critical Jobs task:
 - a. In the Enter Task Information panel, specify the task name and select the engine.
 - b. In the General Filter panel, specify DB* as **Job Name** and set a **Risk Level** different from None as filter criteria, because you want to monitor the critical jobs that risk missing their deadlines.

- c. Click **Finish** to complete the task, leaving the default values in the remaining panels.
3. In the displayed panel, select from the list the task you created and click **Run**. The task output shows that the **Risk Level** for DBSTART job is **Potential Risk**.
4. Select the DBSTART job and click **Critical Path** to view the path of the DBSTART predecessors with the least slack time. The **Critical Path** view does not show any cause for the delay, because no problems occurred for any of the DBSTART predecessors in the critical path. Return to the **Monitor Critical Jobs** task output.
5. Click **Hot List** or the **Potential Risk** hyperlink to get a list of any critical job predecessor that is late, has been running for too long, or has ended with an error. The returned **Hot List** shows DBMAINT as a late job. It is scheduled to run on the CPU2 workstation.
 - a. Click the CPU2 hyperlink.
 - b. After verifying that CPU2 is offline, activate the workstation. The DBMAINT job starts to run.
6. Refresh the **Monitor Critical Jobs** task output. It shows that the **Risk Level** for DBSTART job is now **No Risk**.

Related concepts:

"Workload service assurance" on page 68

Related tasks:

"Creating a task to Monitor Critical Jobs" on page 134

Related reference:

"Limit the number of objects retrieved by queries" on page 190

Monitoring jobs running on multiple engines

This scenario describes how you use the Dynamic Workload Console to create a task to simultaneously monitor jobs that run on multiple engines, which can be in mixed distributed and z/OS environments.

Overview

High efficiency batch processing relies on powerful monitoring capabilities. The need for a single operator to monitor systems is continuously increasing. Up until about 10 years ago, only a limited amount of workload was monitored, but this is increasing to the monitoring of an entire division, and even to an entire company.

Today operators frequently have to monitor multiple large divisions or multiple companies for service providers. These operators work in shifts in multiple geographical locations according to a "follow-the-sun" approach, in some cases. They must try to balance what must be monitored with the size of the monitored environment.

Business scenario

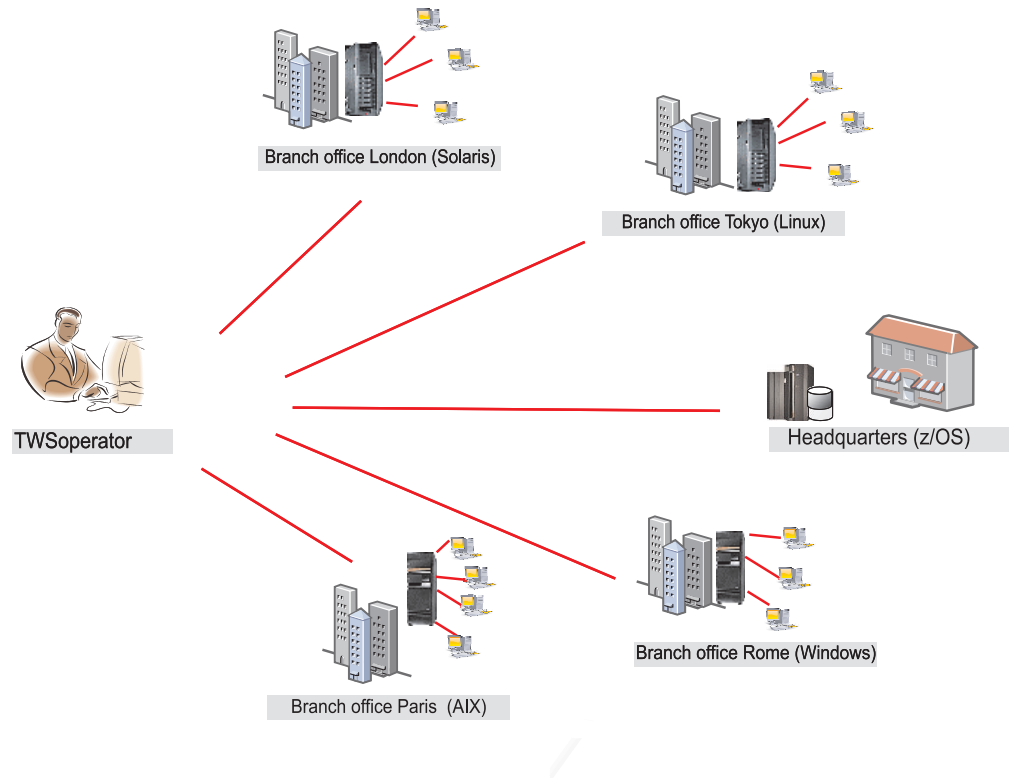
In this scenario, an insurance company, named *Starbank*, consists of a headquarters where its central office accounting department is located, and multiple branch offices located all over the world, where several administrative departments perform accounting activities.

The central office is in charge of the company's entire accounting workload. Therefore, the Tivoli Workload Scheduler operator must verify that all the

workload processing for the *Starbank* company proceeds smoothly and without errors and needs a comprehensive workload management solution.

To achieve this goal, the operator needs to create a task that he can run every day to monitor all the administrative jobs, detecting in real time any possible failures.

However, although the sales department of the company runs its jobs in a z/OS environment, the single business units run their jobs in distributed environments. The operator needs a single console panel, from which he can control all the jobs, both z/OS and distributed, at the same time.



The operator creates a task to monitor jobs that run on multiple engines, including both the environments. He does this by creating and running a task to Monitor Jobs on multiple engines.

Creating a Monitor Jobs task for multiple engines

The operator logs in to the Dynamic Workload Console and, from the navigation



bar on the left, he clicks **System Status and Health > Workload Monitoring > Monitor Jobs > New**.

In the Create Task panel, the operator selects **Monitor Tasks on Multiple Engines > Monitor Jobs** and clicks Next.

Selecting the engines

In the Enter Task Information panel, the operator specifies a name for the task, for example *AccError*, and defines the scheduler engines on which to run the task.

According to a company naming convention policy, all the engine names have a prefix specifying to which department they belong. Therefore, the operator includes in the **Selected Engines** list all the engines named *acc_**. The operator then organizes the list by importance, placing the engines belonging to the most critical company departments (like *Finance* and *Sales*) at the beginning of the list, so as to have their results displayed as the first rows of the table. The task runs following the engine sequence, but the results are displayed altogether, only after the task has run on all the engines in the list.

Defining the filter

In the General Filter panel, the Tivoli Workload Scheduler operator specifies some filtering criteria to limit the results retrieved by the query. Here he starts refining the scope of the query by also considering the amount of information to retrieve. Defining a meaningful filter is very important to avoid unnecessary overhead, considering that the task runs on multiple engines. First, the operator sets the automatic refresh time to *600* so as to receive the updated monitoring results every 600 seconds (10 minutes). He then filters for jobs based on their job streams. According to a company policy, all administrative job streams begin with the company name followed by the department code. In our scenario, the operator looks for all the job streams whose identifier starts with *Starb** that did not complete successfully.

Selecting the columns

In the Columns Definition panel, the operator selects the information to display in the table containing the query results. According to the columns he chooses, the corresponding information is displayed in the task results table. In our scenario, for all the jobs resulting from the query, the operator wants to see their statuses, the job streams they belong to, when they were scheduled to run, and the engines on which they ran. Then, if more details are necessary, he can drill down into this information displayed in the table of results and navigate it.

Results

In the All Configured Tasks panel, the operator can see the main details about the task that he has just created and launch the task immediately. The task is now in the list of saved tasks from where the operator can open and modify it any time. To find the task from the displayed task lists, he clicks the following options:



System Status and Health > Workload Monitoring > Monitor Jobs.

The operator has created a task that can be run every day to highlight possible critical failure in real time. If there is a failure in any of the administrative jobs run by the selected offices, the operator discovers it no later than 10 minutes after the error occurs.

Running the Monitor Jobs task for multiple engines



To launch the task, the operator clicks **System Status and Health > Workload Monitoring > Monitor Jobs**.

The operator clicks *AccError* task to launch it. Because some engine connections do not work correctly, the Checking engine connections panel reports some errors on two of the eight engines defined. The failing connections are the *Tokyo* and *Paris* offices. The operator could ignore the failed connections and proceed, running the task on the successful engines only. However, monitoring the entire workload running in all the branch offices is crucial to his activity, and he does not want to skip any engine connection. Therefore, by clicking **Fix it** next to each failing engine connection, the operator opens a dialog where he can enter the credentials required for that engine. After entering the correct credentials, also the remaining engine connections work successfully and the operator clicks **Proceed** to run the task against all the engines.

Viewing results and taking corrective actions


Viewing the results of the *AccError* task, the operator realizes that there is a job in error, named *PayAcc1*. He right-clicks the job to open its job log, to better determine the cause and effects of this error.

From the job log, he finds out that only the last step of the job failed, which was a data backup process. This step can be done manually at a later time. The most important part of the job, consisting of the accounting processes related to payrolls, completed successfully.

Now the operator needs to determine the impact that this job in error has on the overall plan. To do this, he selects the *PayAcc1* job and clicks **Graphical View > Impact View**. From this view, he realizes that this job is a predecessor dependency of another job, named *Balance1*. The operator releases the failing job dependency so as to make it possible for the successor *Balance1* to start and the whole workload processing to complete.

A second job in error results from the *AccError* task. It is a z/OS job, named *Info*. The operator selects this job from the list and right-clicks it to open the **Operator Instructions** that give him important information about what to do. According to the instructions, this is an optional procedure, which can be skipped without consequence for the entire processing. Therefore, the operator right-clicks the job and cancels it.

The operator then refreshes the view to ensure that there are no other jobs in error.

To view connection status information and statistical information about the engines against which the task was run, the operator clicks the statistical icon  on the table toolbar.

A pie chart showing the number of query results and job status is displayed for each engine on which the task ran successfully. By clicking the pie sections, he can see further details. If the task did not run successfully on one or more engines, he sees a message containing details about the errors.

Related tasks:

“Creating a task to Monitor Tasks on Multiple Engines” on page 137

Related reference:

“Limit the number of objects retrieved by queries” on page 190

Chapter 16. Troubleshooting the Dynamic Workload Console

Accessing troubleshooting information.

You can find information about how to troubleshoot problems with the Dynamic Workload Console, related to connections, performance, user access, reports, and others at the following link: *Tivoli Workload Scheduler Troubleshooting*, under the section about troubleshooting Dynamic Workload Console problems.

Note: If you print PDF publications on other than letter-sized paper, set the option in the File -> Print window that enables Adobe Reader to print letter-sized pages on your local paper.

Chapter 17. Reference

This section provides some reference information that can be useful to perform the main tasks and activities from the Dynamic Workload Console.

Accessing online publications

Accessing the Tivoli products online publications in the Tivoli software information center website.

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli software information center website. Access the Tivoli software information center at the following links:

- Tivoli Workload Scheduler Information Center, to access all the publications related to the Tivoli Workload Scheduler product
- Tivoli Workload Scheduler Wiki Media Gallery, to access demos (only available in English) about how to use the Tivoli Workload Scheduler product
- *Tivoli Workload Automation wiki*, to access information about Tivoli Workload Scheduler such as best practices, product features, and new tools.

Note: If you print PDF publications on other than letter-sized paper, set the option in the File -> Print window that enables Adobe Reader to print letter-sized pages on your local paper.

Dynamic Workload Console global settings

A copy of this file is installed locally in the following path after you install the Dynamic Workload Console:

`<JazzSM_profile_dir>/profile/registry`

You can also find a copy of this file, in template form, on the installation DVD in the directory `/utilities/TdwcGlobalSettings.xml`.

Modify the file replacing default values with customized ones and enabling commented sections, and then save the file to the `<JazzSM_profile_dir>/registry` directory. The complete path of the file is as follows:

For Windows:

`C:\Program Files\IBM\JazzSM\profile\registry\TdwcGlobalSettings.xml`

For UNIX/Linux:

`/opt/ibm/JazzSM/profile/registry/TdwcGlobalSettings.xml`

Role based customization

The file is organized into several sections that group similar properties. Sections can be repeated multiple times in the same file and applied differently to different user roles.

To apply a section only to the users belonging to a role, the section must be included within the following setting:

settings role

The user for which the following configuration must be applied. Default value: all users, unless otherwise specified.

Only one **settings** section can be specified for each role. If a user has more than one role, the settings associated to the higher role are taken into consideration.

Example:

```
<?xml version="1.0"?>
<tdwc>
  <settings>
    <graphViews>
      <property name="planViewNewWindow" value="true"/>
    </graphViews>
  </settings>
  <settings role="TWSWEBUIOperator">
    <graphViews>
      <property name="planViewNewWindow" value="false"/>
    </graphViews>
  </settings>
</tdwc>
```

Override graphical view limits

This section contains the configuration parameters that apply to the graphical views in the plan, such as the maximum number of objects shown in each view.

planViewMaxJobstreams

The maximum number of job streams displayed in the Plan View. Default value is **1000**.

jobstreamViewLimit

The maximum number of objects displayed in the Job Stream View. Default value is **1000**.

impactViewLimit

The maximum number of job streams displayed in the Impact View. Default value is **2000**.

preProdPlanViewMaxJobstreams

The maximum number of job streams displayed in the preproduction plan view. Default value is **1000**.

```
<?xml version="1.0"?>
<tdwc>
.
.
  <settings>
    <graphViews>
      <property name="planViewMaxJobstreams" value="1000"></property>
      <property name="jobstreamViewLimit" value="1000"></property>
      <property name="impactViewLimit" value="1000"></property>
      <property name="preProdPlanViewMaxJobstreams" value="1000"></property>
    </graphViews>
  </settings>.
.
</tdwc>
```

See “TdwcGlobalSettings.xml sample” on page 194 to view the complete syntax for the file.

Plan View in new window

This section is used to prevent Internet Explorer 7 from freezing while using the Plan View. To solve the problem, set value to **true**.

planViewNewWindow

Set it to **true** if you want the plan view to be displayed in a new window each time it is launched. Default value is **false**.

```
<?xml version="1.0"?>
<tdwc>
.
.
  <settings>
<graphViews>
<property name="planViewNewWindow" value="true"/>
</graphViews>
.
.
  </settings>
</tdwc>
```

See “TdwcGlobalSettings.xml sample” on page 194 to view the complete syntax for the file.

Disable and customize NewsFeed function

This section contains the configuration details to be constantly up-to-date with product information.

FeedURL

Contains the URL from which you receive news and updates. Default value is: <https://www.ibm.com/developerworks/community/wikis/form/anonymous/api/wiki/585f5525-a7f5-48ef-9222-50ad582e85f4/page/e599dd3c-8dc3-4ab6-89fd-33f81a994799/attachment/de677e63-5a9d-46db-a010-18ca38f05812/media/tws.jsonp>

FeedType

A string that identifies the format of update information. Default value is **JSONP**.

PollInterval

The interval in seconds between two checks for updates. Default value is **600**.

PollInitialDelay

An initial delay in seconds before the first attempt to read the news feeds. After the initial load, the poll interval is used. Default value is **120**.

NewsFeed

Property used to add further customized news feeds. Specify the format and address of the file that contains the customized communication. Supported formats are RSS 2.0 and ATOM 1.0. You must write the communication in ATOM 1.0 or RSS 2.0 format and store this file in the an HTTP server complying with the *same origin policy*. For browser security reasons, this policy permits to access information only on server using the same protocol, hostname and port number as the one to which you are connected. Optionally, if you want to store your customized feed on an external server, you must configure an HTTP reverse proxy server mapping the external server address.

```
<property name="NewsFeed" type="RSS"
value="http://DWC_hostname:portnumber.com/news.rss" />
```

Note: To specify multiple feeds, you must specify multiple **NewsFeed** properties.

NewsFeedCategory

The name of the customized information. It can be used to identify informational, warning or alert messages, for example. The path to an image can also be added to better identify the information with an icon.

To add more category images, specify a list of properties named **NewsFeedCategory**, for example:

```
<property name="NewsFeedCategory" value="my company info"
icon="http://www.my.company.com/info.png" />
<property name="NewsFeedCategory" value="my company alert"
icon="http://www.my.company.com/alert.png" />
```

If no customized feed is specified, the default feed is used, which retrieves the latest product information from official support sites. To disable any notification, comment the entire section. To disable only external notifications about product information updates, assign an empty string as value to the FeedURL property of JSONP feed like:

```
<property name="FeedURL" type="JSONP" value="" />
```

Example:

```
<?xml version="1.0"?>
<tdwc>
.
.
  <settings>
<NewsFeed>
<property name="NewsFeed" type="RSS"
value="http://www.DWC_hostname:portnumber.com/my_rss.xml" />
<property name="NewsFeed" type="ATOM"
value="http://www.DWC_hostname:portnumber.com/my_atom.xml" />

<property name="PollInterval" value="600" />
<property name="PollInitialDelay" value="1" />

<property name="FeedURL" type="JSONP" value="" />

<property name="NewsFeedCategory"
value="my company info" icon="http://www.DWC_hostname:portnumber.com
/info.png" />
<property name="NewsFeedCategory"
value="my company alert" icon="http://www.DWC_hostname:portnumber.com
/alert.png" />

</NewsFeed>
</settings>
.
.
</tdwc>
```

See “TdwcGlobalSettings.xml sample” on page 194 to view the complete syntax for the file.

Related tasks:

“Customizing the list of notification recipients” on page 25
How to customize the list of recipients of the news notification beacon, by specifying who must receive what kind of notification or communication.

“Disabling news notification” on page 24
How to enable and disable news notification preventing the beacon from

appearing on your screen when an update for the product is made available.

“Sending internal communications” on page 26

How to use the news feed beacon to transmit internal communications to all or some Dynamic Workload Console users.

Disable and customize the creation of predefined tasks

This section defines the environment for which predefined tasks are created.

precannedTaskCreation

Some predefined tasks are created by default and are available when you log in to the console. There is a predefined Monitor task for every object, for both z/OS and distributed engines. Default value is **all**. To change this setting, use one of the following values:

all All predefined tasks are created. This is the default.

distributed

Only predefined tasks for distributed engines are created

zos Only predefined tasks for z/OS engines are created

none No predefined task is created.

```
<?xml version="1.0"?>
<tdwc>
.
.
  <settings>
<application>
<property name="precannedTaskCreation" value="all"/>
</application>
  </settings>
.
.
</tdwc>
```

See “TdwcGlobalSettings.xml sample” on page 194 to view the complete syntax for the file.

Add customized URL to job and job streams

This section contains URLs where you can store customized documentation about your jobs or job streams. By default, this setting is not specified. If you want to associate customized documentation to a job or job stream, use this setting to specify the external address where this information is located.

If you want to specify a URL where customized documentation for a job and job stream is stored, uncomment the section lines, specify the required URL, and optionally assign a name to the UI label by specifying a value for the customActionLabel property. By default this name is **Open Documentation**. This label is then displayed in the **More Actions** menus in Monitor Jobs and Monitor Job Streams tasks, as well as in the graphical views of the plan (in the object's tooltips, context menus and properties). In this example, selecting **Open Documentation** accesses the relevant documentation making it possible to open the documentation while monitoring your job or job stream in the plan.

To implement this setting, assign values to the following keywords:

customActionLabel

The name of the action displayed in menus, object properties, and tooltips

to access customized documentation about your jobs or job streams. By default this name is "Open Documentation" unless you customize the name with this keyword.

jobUrlTemplate

The address of your job documentation. No default value available.

jobstreamUrlTemplate

The address of your job stream documentation. No default value available.

```
<?xml version="1.0"?>
<tdwc>
.
.
<settings>
  <twsobjectDoc>
    <property name="jobstreamUrlTemplate"
      value="http://www.yourhost.com/tws/docs/jobstream/${js_name_w}"/>
    <property name="jobUrlTemplate"
      value="http://www.yourhost.com/docs/jobs/${job_name_w}"/>
    <property name="customActionLabel" value="Your Custom Label Name"/>
  </twsobjectDoc>
</settings>
.
.
</tdwc>
```

See “TdwcGlobalSettings.xml sample” on page 194 to view the complete syntax for the file.

These properties must be valid URLs, containing one or more of the variables listed in the table below.

If you use any of the following special characters in the URL, you must write them as follows:

Table 13. Syntax for special characters

Special characters	Write them as...
<i>quote (")</i>	\"
<i>apostrophe (')</i>	'
<i>ampersand (&)</i>	&
<i>less than (<)</i>	<
<i>greater than (>)</i>	>
<i>backslash (\)</i>	\\

Multiple variables can be included in a URL and must be specified using the following syntax: `${variable}`:

Table 14. Variables used in the URL definition

Name	Object	Description
job_number_w	Job z/OS	The number of the job
job_wkst_w	Job	The name of the workstation on which the job runs
job_jsname_w	Job	The name of the job stream that contains the job

Table 14. Variables used in the URL definition (continued)

Name	Object	Description
job_jswkst_w	Job	The name of the workstation on which the job stream runs
job_actualarrival_w	Job z/OS	The actual start time of the job (date format: YYYY-MM-DDThh:mm:ss)
job_actualend_w	Job z/OS	When the job actually completed (date format: YYYY-MM-DDThh:mm:ss)
job_starttime_w	Job	The start time of the job (date format: YYYY-MM-DDThh:mm:ss)
job_id_w	Job	The ID of the job
job_returncode_w	Job	The return code of the job
js_name_w	Job stream	The name of the job stream that contains the job
js_wkst_w	Job stream	The name of the workstation on which the job stream runs
js_id_w	Job stream	The job stream ID
js_latest_start_w	Job stream	The latest time at which a job stream can start (date format: YYYY-MM-DDThh:mm:ss)
engine_name_w	Engine	The name of the engine connection
engine_host_w	Engine	The hostname of the engine connection
engine_port_w	Engine	The port number of the engine connection
engine_plan_w	Engine	The ID of selected plan
engine_serv_w	Engine	The remote server name of the engine connection

User registry

Use this section to configure some properties related to the User Registry in use.

groupIdMap

The property is related to the groups of User Registry, and can be modified to map and display the specified value of each group. The default is the common name of the group.

Examples:

```
<?xml version="1.0"?>
<tdwc>
.
.
<settings>
<security>
<property name="groupIdMap" value="cn"></property>
</security>
```

```

</settings>
.
.
</tdwc>

```

Therefore, if you need to change the default value "cn" to "racfid", you can define this property as follows:

```
<property name="groupIdMap" value="racfid"></property>
```

See "TdwcGlobalSettings.xml sample" on page 194 to view the complete syntax for the file.

z/OS http connections

Use this section to configure the timeout to read and write information on Tivoli Workload Scheduler for z/OS engine. When you connect to the Tivoli Workload Scheduler for z/OS engine to retrieve a list of defined objects, you receive an error message if the list is not returned within the timeout period. The value is expressed in milliseconds.

Example:

```

<?xml version="1.0"?>
<tdwc>
.
.
  <settings>
<http>
<property name="zosHttpTimeout" value="90000" />
</http>
.
.
  </settings>
</tdwc>

```

See "TdwcGlobalSettings.xml sample" on page 194 to view the complete syntax for the file.

Limit the number of objects retrieved by queries

If you are connected to engines V9.1 this setting is ignored.

Use this section to configure: the number of results displayed for Monitor tasks, the maximum number of rows to display on each page, and the number of direct queries to maintain in history.

If you want to limit the number of results produced by your queries, you can specify the maximum number of items that must be retrieved using the monitorMaxObjectsPM property. The minimum number of retrieved results is 500.

The default value is -1; any value lower than 0 means that there is no limit in the number of objects retrieved.

Because data is extracted in blocks of 250 rows, the value you enter is adjusted to complete an entire block. For example, if you specify a limit of 500, only 500 elements are retrieved, while if you specify a limit of 600, 750 elements are retrieved.

For Multiple engine tasks, this limit is applied to each engine included in the query. Therefore, if you specify a limit of 500 results and, for example, you run a

Monitor jobs on multiple engine task on three engines, the results produced by your query will be no more than 500 *for each engine*, for a maximum of 1500 rows.

Note: This setting does not apply to Monitor critical jobs tasks.

To set the maximum number of rows to display in a table view, configure the `maxRowsToDisplay` property.

To set the maximum number of direct queries to maintain in history, configure the `maxHistoryCount` property. These queries are available from the pull-down for the Query field on the Direct Query page.

```
<?xml version="1.0"?>
<tdwc>
.
.
<settings>
  <monitor>
    <property name="monitorMaxObjectsPM" value="2000"></property>
  </monitor>

  <monitor>
    <property name="maxRowsToDisplay" value="25"></property>
  </monitor>

  <monitor>
    <property name="maxHistoryCount" value="100"></property>
  </monitor>
</settings>
.
.
</tdwc>
```

See “TdwcGlobalSettings.xml sample” on page 194 to view the complete syntax for the file.

Related concepts:

“Job” on page 38

“Job stream” on page 38

“Workload service assurance” on page 68

“Monitoring jobs running on multiple engines” on page 175

Related tasks:

“Creating a task to Monitor Jobs” on page 132

“Creating a task to Monitor Critical Jobs” on page 134

“Creating a task to Monitor Tasks on Multiple Engines” on page 137

“Creating a task to Monitor Job Streams” on page 138

“Creating a task to Monitor Job Streams on Multiple Engines” on page 140

“Creating job stream definitions” on page 83

Related information:

“Using workload service assurance to monitor z/OS critical jobs” on page 173

Limit task and engine sharing

Use this section to prevent users from sharing tasks and engines.

By default there is no limit to task and engine sharing and all users are authorized to share their tasks and engine connections. If you want to change this behavior, preventing users from sharing tasks and engines, set this property to **true**.

The property default value is **false**, set it to **true** to enable the limit:

limitShareTask

Set to true to prevent users from sharing tasks.

limitShareEngine

Set to true to prevent users from sharing engine connections.

```
<?xml version="1.0"?>
<tdwc>
.
.
<settings>
  <security>
    <property name="limitShareTask" value="false" />
    <property name="limitShareEngine" value="false" />
  </security>
</settings>
.
.
</tdwc>
```

See “TdwGlobalSettings.xml sample” on page 194 to view the complete syntax for the file.

Entries in Workload Designer search windows

This section contains the configuration parameters that apply to the search views of Workload Designer.

search_max_limit

This optional parameter sets the maximum number of entries displayed in Workload Designer search windows. The default value is 250. It is recommended not to use values greater than 999.

Example:

```
<?xml version="1.0"?>
<tdwc>
.
.
<settings>
  <search>
    <property name="search_max_limit" value="500"></property>
  </search>
</settings>
.
.
</tdwc>
```

Show all dependencies

This section defines whether to show all dependencies displayed, regardless of their being satisfied or not.

ShowDependencies

When you open the dependencies panel from Monitor jobs and Monitor job streams task results, by default only **Not Satisfied** dependencies are shown. Uncomment this section and leave the value set to "**true**" to have all dependencies displayed, regardless of their being satisfied or not. Possible values are:

true All dependencies displayed, regardless of their being satisfied or not.

false Only not satisfied dependencies are displayed.

```
<?xml version="1.0"?>
<tdwc>

.
.
<settings>
  <ShowDependencies>
    <property name = "AlwaysShowAllDependencies"
              value="true"></property>
  </ShowDependencies>
</settings>
.
.
</tdwc>
```

See “TdwGlobalSettings.xml sample” on page 194 to view the complete syntax for the file.

Auditing mobile app activity

This section defines whether to track activities performed in the Self-Service Catalog and Self-Service Dashboards applications in an auditing log file.

For information about the name and location of the log file, see Log file for the Self-Service mobile applications.

For information about the name and location of the log file, see the logs and traces section in the *Troubleshooting Guide*.

SSAuditing

This value is set to **"true"** by default so that operations performed in the Self-Service Catalog and Self-Service Dashboards applications are written to a log file. The log file contains information such as creation, modification and deletion dates, the operations performed in the mobile apps, and the user performing the operations. Possible values are:

true Operations performed in the Self-Service Catalog and Self-Service Dashboards applications are tracked in an auditing log file.

false Operations performed in the Self-Service Catalog and Self-Service Dashboards applications are not tracked in an auditing log file.

SSAuditingLogSize

The maximum size of a log file in KB. When a log file reaches the maximum size, the system rolls that log file over and creates a new file. By default, the maximum size of a log file is 100 KB.

SSAuditingLogFiles

The default number of log files to create. When this number is met and the latest log file reaches its maximum size, the system deletes the oldest log file and rolls the latest file over and creates a new file.

```
<?xml version="1.0"?>
<tdwc>
.
.
<settings>
  <SSCAuditing>
    <property name = "SSAuditing"           value="true"></property>
    <property name = "SSAuditingLogSize"    value="100"></property>
    <property name = "SSAuditingLogFiles"   value="2"></property>
```


that a new action, Open Documentation, is inserted in the More Actions menu for Monitor Jobs and Monitor Job Streams tasks. The new action links to the specified URL

You can customize the URL template by using variables. The variables have the syntax `${<variable_name>}`

For the complete list of variables, see the documentation.

```
-->
<!--
<twsoObjectDoc>
<property name="jobstreamUrlTemplate" value="http://www.yourhost.com/tws/docs/jobstream/${js_name_w}" />
<property name="jobUrlTemplate" value="http://www.yourhost.com/docs/jobs/${job_name_w}" />
<property name="customActionLabel" value="Custom Action" />
</twsoObjectDoc>
-->

<!--
##### SECTION 6 - USER REGISTRY #####
In this section you can configure some properties about the User Registry in use.
The property groupIdMap is related to the groups of User Registry, and can be modified
to map and display the specified value of each group. By default the common name of the
group is displayed.
-->
<!--
<security>
<property name="groupIdMap" value="cn"></property>
</security>
-->
<!--
##### SECTION 7 - Z/OS HTTP CONNECTIONS #####
Use this section for increase or decrease timeout for http connection in Z/OS
environment. Change this setting if you receive a connection timeout using plugin
actions/picklists.

The setting is in milliseconds.
-->
<!--
<http>
<property name="zosHttpTimeout" value="90000" />
</http>
-->
<!--
##### SECTION 8 - LIMIT THE NUMBER OF OBJECTS RETURNED BY THE QUERIES #####
Use this section to configure: the number of results displayed for Monitor tasks, the
maximum number of rows to display on each page, and the number of direct queries to
maintain in history. This setting applies to all tasks except for Monitor critical jobs
and Monitor jobs on multiple engines.
If you want to limit the number of results produced by your queries, you can specify the
maximum number of items that must be retrieved. The default value is -1; any value lower
than 0 means that there is no limit in the number of objects retrieved. The minimum number
of retrieved results is 500. Because data is extracted in blocks of 250 rows, the value
you enter is adjusted to complete an entire block. For example, if you specify a limit of
500, only 500 elements are retrieved, while if you specify a limit of 600, 750 elements
are retrieved.
To set the maximum number of rows to display in a table view, configure the
maxRowsToDisplay property.
To set the maximum number of direct queries to maintain in history, configure the
maxHistoryCount property. These queries are available from the pull-down for the Query
field on the Direct Query page.

<monitor>
  <property name="monitorMaxObjectsPM" value="2000"></property>
</monitor>

<monitor>
  <property name="maxRowsToDisplay" value="25"></property>
</monitor>

<monitor>
  <property name="maxHistoryCount" value="100"></property>
</monitor>
-->

<!--
##### SECTION 9 - LIMIT TASK AND ENGINE SHARING #####
Use this section to prevent users from sharing tasks and engines.
By default there is no limit to task and engine sharing and all users are authorized to share
their tasks and engine connections. If you want to change this behavior, preventing users from
sharing tasks and engines, set this property to true. The property default value is false,
set it to true to enable the limit:

-->
<!--
<security>
<property name="limitShareTask" value="false" />
<property name="limitShareEngine" value="false" />
</security>
-->

<!--
```

```
#####
##### SECTION 10 - CHANGE DEFAULT BEHAVIOR FOR DEPENDENCIES PANEL #####
#####

Use this section to change the default behavior of the UI when displaying dependencies in
the dependencies panel. By setting this value to true, by default, all dependencies are
displayed, and not just the unsatisfied ones.
-->
<!--
<ShowDependencies>
  <property name = "AlwaysShowAllDependencies"
value="true"></property>
</ShowDependencies>
-->

#####
##### SECTION 11 - CHANGE DEFAULT BEHAVIOR FOR SSC AND SSD AUDITING #####
#####

Use this section to change the default behavior of the auditing of activities performed
using the Self-Service Catalog and the Self-Service Dashboards applications. By default,
auditing is enabled. You can also set the maximum size of the log file before it rolls
over to a new log file, and the maximum number of log files maintained.
-->
<!-- <SSCAuditing>
  <property name = "SSAuditing" value="true"></property>
  <property name = "SSAuditingLogSize" value="100"></property>
  <property name = "SSAuditingLogFiles" value="2"></property>
-->

-->

</settings>

<!--
#####
##### SETTINGS FOR ALL TWSWEBUIAdministrators users #####
#####
-->
<settings role="TWSWEBUIAdministrator">
<!-- Put here setting to be applied only to users with TWSWEBUIAdministrator role -->
</settings>
<!--
#####
##### SETTINGS FOR ALL TWSWEBUIOperators users #####
#####
-->
<settings role="TWSWEBUIOperator">
</settings>
<!--
#####
##### SETTINGS FOR ALL TWSWEBUIConfigurator users #####
#####
-->
<settings role="TWSWEBUIConfigurator">
</settings>
<!--
#####
##### SETTINGS FOR ALL TWSWEBUIDeveloper users #####
#####
-->
<settings role="TWSWEBUIDeveloper">
</settings>
<!--
#####
##### SETTINGS FOR ALL TWSWEBUIAnalyst users #####
#####
-->
<settings role="TWSWEBUIAnalyst">
</settings>

</tdwc>
```

Users and groups

The Dynamic Workload Console uses the Dashboard Application Services Hub capabilities for defining users and to authorize them to view and use items on the navigation menu. Users are granted access to resources based on the role to which they have been assigned. The groups in which users are defined or the role assigned to users determines the operations they can perform and which resources are visible to them. This means that depending on your user role designation, you might not see all of the items described in this help system. The groups or predefined roles available are:

TWSWEBUIAdministrator

Users in this group can see the entire portfolio and use all features of the Dynamic Workload Console.

Users in this group can also access and use all features of the Self-Service Catalog and the Self-Service Dashboards mobile applications. From the Self-Service Catalog mobile application, these users can create and edit catalogs, create and edit services, add services to catalogs, submit services

associated to job streams, and share catalogs and services with other users. From the Self-Service Dashboards mobile application, these users can create and edit dashboards to filter for jobs and workstations, display a dashboard of results, perform recovery actions on a single result.

TWSWEBUIConfigurator

Users in this group can manage Dynamic Workload Console scheduler connections, user preferences, and scheduling environment design.

TWSWEBUIOperator

Users in this group can see Dynamic Workload Console:

- All Monitor tasks
- Jobs and job streams to be submitted on request
- Set User Preferences

TWSWEBUIDeveloper

Users in this group can create, list, and edit workload definitions, workstations, and event rule definitions in the Tivoli Workload Scheduler database.

TWSWEBUIAnalyst

Users in this group can manage Dynamic Workload Console reports and user preferences.

Users in this group can also access the Self-Service Catalog and the Self-Service Dashboards mobile applications but the actions they can perform are limited to submitting service requests (job streams) from the Self-Service Catalog and, from the Self-Service Dashboards mobile application, displaying a dashboard of results and performing recovery actions on them.

TWSWEBUIBusinessDeveloper

Users in this group can access and use the Self-Service Catalog and the Self-Service Dashboards mobile applications. From the Self-Service Catalog mobile application, these users can create and edit catalogs, create and edit services, add services to catalogs, delete services and catalogs, and submit services associated to job streams. From the Self-Service Dashboards mobile application, these users can create and edit dashboards to filter for jobs and workstations, display and view a dashboard of results, delete dashboards, and perform recovery actions on a single result. To share catalogs, services, and dashboards with other users, the TWSWEBUIBusinessDeveloper can assign them to the custom roles that the TWSWEBUIBusinessDeveloper possesses but not to predefined roles. Users with these same custom roles can work with the catalogs, services, and dashboards. Users with all of the custom roles can submit services; view, edit, and delete services, catalogs, and dashboards; but users with only one or some of the custom roles can only submit services, and view services, catalogs, and dashboards.

If a user with the Administrator role, creates catalogs, services, and dashboards and does not assign any roles to them, then users with the TWSWEBUIBusinessDeveloper role cannot see or work with them.

Note: If a custom role is removed from a catalog, service, or dashboards, in addition to the TWSWEBUIBusinessDeveloper user, users with this same custom role can no longer see and work with them, even if they possess other custom roles that are currently assigned to the catalog or service. The Administrator must reassign the custom role to the catalog, service, or

dashboard to make it accessible again to the TWSWEBUIBusinessDeveloper user and other users with the same custom role.

The following table lists some entries of the navigation toolbar, and some activities you can perform from the Dynamic Workload Console. Beside each item, the table shows the groups whose users are authorized to access them.

Table 15. Menu and Group Permissions

Menu Item	Groups with Permission
Quick Start	TWSWEBUIAdministrator
All Configured Tasks	TWSWEBUIAdministrator TWSWEBUIOperator
Manage Workload Reports	TWSWEBUIAdministrator TWSWEBUIAnalyst
Administration -> Workload Design	TWSWEBUIAdministrator TWSWEBUIDeveloper
Administration -> Workload Forecast	TWSWEBUIAdministrator TWSWEBUIOperator
Administration -> Workload Submission	TWSWEBUIAdministrator TWSWEBUIOperator
Administration -> Monitor	TWSWEBUIAdministrator TWSWEBUIOperator
Administration -> Workload Design	TWSWEBUIAdministrator TWSWEBUIConfigurator
Administration -> Monitor	TWSWEBUIAdministrator TWSWEBUIOperator
Workload Reporting	TWSWEBUIAdministrator TWSWEBUIAnalyst
System Configuration ->Manage Engines	TWSWEBUIAdministrator TWSWEBUIConfigurator
System Configuration -> Set User Preferences	TWSWEBUIAdministrator TWSWEBUIOperator TWSWEBUIConfigurator TWSWEBUIDeveloper TWSWEBUIAnalyst
System Configuration -> Manage Settings	TWSWEBUIAdministrator

Type of communication based on SSL communication options

Based on the authentication types that you defined for the workstations in your network, communication between workstations is different. The following table summarizes the types of connection for the different authentication type settings.

Table 16. Type of communication based on workstation SSL communication options

Fault-tolerant Agent (Domain Manager)	Domain Manager (Parent Domain Manager)	Connection Type
Disabled	Disabled	TCP/IP
Allow Incoming	Disabled	TCP/IP
Upward Forced	Disabled	No connection

Table 16. Type of communication based on workstation SSL communication options (continued)

Fault-tolerant Agent (Domain Manager)	Domain Manager (Parent Domain Manager)	Connection Type
All Forced	Disabled	No connection
Disabled	Upward Forced	TCP/IP
Allow Incoming	Upward Forced	TCP/IP
Upward Forced	Upward Forced	SSL
All Forced	Upward Forced	SSL
Disabled	Allow Incoming	TCP/IP
Allow Incoming	Allow Incoming	TCP/IP
Upward Forced	Allow Incoming	SSL
All Forced	Allow Incoming	SSL
Disabled	All Forced	No connection
Allow Incoming	All Forced	SSL
Upward Forced	All Forced	SSL
All Forced	All Forced	SSL

For details about how to create SSL certificates and how to set local options for SSL communication, see *Tivoli Workload Scheduler Administration*.

Related tasks:

“Creating distributed workstations” on page 75

Status description and mapping for distributed jobs

Distributed

There are the following types of status for distributed jobs:

“Job status”

A subset of internal status that is common for both Tivoli Workload Scheduler for distributed and Tivoli Workload Scheduler for z/OS environments.

“Job internal status” on page 200

The Tivoli Workload Scheduler job status registered on the workstation where the job is running. The internal status uniquely identifies a job status in Tivoli Workload Scheduler.

Job status

Table 17 lists the job statuses.

Table 17. Job status

This job status ...	Means that ...
Waiting	The job is waiting for its dependencies to be resolved.
Ready	The dependencies of the job have been resolved and the job is ready to run.
Running	The job is running.

Table 17. Job status (continued)

This job status ...	Means that ...
Successful	The job completed successfully.
Error	The job stopped running with an error.
Canceled	The job was canceled.
Held	The job was put on hold.
Undecided	The job status is currently being checked.
Blocked	The job was blocked because of unfulfilled dependencies.

Job internal status

Table 18 lists the job internal statuses.

Note: The + flag written beside the INTRO and EXEC statuses means that the job is managed by the local **batchman** process.

Table 18. Job internal status

This job status ...	Means that ...
ABEND	The job terminated with a nonzero exit code or with an exit code outside the defined RC mapping.
ABEND P	An ABEND confirmation was received, but the job is not completed.
ADD	The job is being submitted.
BOUND	For shadow jobs, it means that the shadow job matched a remote job instance in the remote plan. For TWS for z/OS agents, it means that the job is on the JES queue.
CANCEL	The job was canceled.
CANCEL P	The job is pending cancelation. Cancelation is deferred until all of the dependencies, including at time dependencies, are resolved.
DONE	The job completed in an unknown status.
ERROR	For internetwork and cross dependencies only, an error occurred while checking for the remote status.
EXEC	The job is running.
EXTRN	For internetwork dependencies only, the status is unknown. An error occurred, a rerun action was just performed on the job in the external job stream, or the remote job or job stream does not exist.
FAILED	Unable to launch the job.
FENCE	The job's priority is below the fence.
HOLD	The job is awaiting dependency resolution.
INTRO	The job is introduced for launching by the system.
PEND	The job completed, and is awaiting confirmation.
READY	The job is ready to launch, and all dependencies are resolved.
R JOB	The job is running.
SCHED	The job's at time has not arrived.
SUCC	The job completed with an exit code of zero.
SUCC P	A SUCC confirmation was received, but the job is not completed.
SUSP	The job was blocked because of unfulfilled dependencies.

Table 18. Job internal status (continued)

This job status ...	Means that ...
USER STAT	The job was put on hold by the user.
WAIT	The job is waiting to fulfill its dependencies.
WAITD	The job is waiting to fulfill its dependencies.

Job status mapping

Table 19 describes how a job status is mapped to the corresponding job internal status.

Table 19. Job status mapping

This job status ...	Maps to this job internal status
Waiting	ADD, PEND, WAIT, WAITD, INTRO, HOLD
Ready	READY
Running	EXEC, SUCC P, ABEND P, R JOB, BOUND
Successful	SUCC
Error	ABEND, FAILED
Canceled	Status of the job when it was canceled. Canceled flag is set.
Held	Priority = 0, WAIT, READY, USER STAT
Undecided	ERROR, EXTRN
Blocked	SUSP

Related concepts:

“Job” on page 38

Related tasks:

“Creating a task to Monitor Jobs” on page 132

“Creating a task to Monitor Critical Jobs” on page 134

“Creating a task to Monitor Tasks on Multiple Engines” on page 137

Status description and mapping for z/OS jobs

z/OS

There are the following types of status for z/OS jobs:

“z/OS job status”

A subset of internal statuses common for both Tivoli Workload Scheduler distributed and z/OS environments.

“z/OS job internal status” on page 202

The job status registered on the Tivoli Workload Scheduler controller. The internal status uniquely identifies the status of a z/OS job.

z/OS job status

Table 20 on page 202 shows the z/OS job statuses that are displayed by the Dynamic Workload Console.

Table 20. z/OS job status

This job status ...	Means that ...
Waiting	The job is waiting for its dependencies to be resolved.
Ready	The dependencies of the job have been resolved and the job is ready to run.
Running	The job is running.
Successful	The job completed successfully.
Error	The job has stopped running with an error.
Canceled	The job was canceled.
Held	The job was put in hold.
Undefined	The job status is currently being checked.
Suppressed by Condition	The job is suppressed because the condition dependencies associated to its predecessors are false.

z/OS job internal status

Table 21 shows the z/OS job internal statuses that are displayed by the Dynamic Workload Console and how they map to the status displayed on the Tivoli Workload Scheduler for z/OS controller.

Table 21. z/OS job internal status

This job internal status ...	Means that ...	Maps to ...
Arriving	The job is ready for processing; no predecessors were defined.	A
Complete	The job has completed	C
Deleted	The job has been deleted from the plan	D
Error	The job has ended-in-error.	E
Interrupted	The job is interrupted.	I
Ready	The job is ready for processing; all predecessors are complete.	R
Started	The job has started	S
Undefined	The job status is being evaluated.	U
Waiting	The job is waiting for a predecessor to complete.	W
Ready - non-reporting workstation	Ready - At least one predecessor is defined on a <i>nonreporting</i> workstation; all predecessors are complete.	*
Suppressed by Condition	The condition dependencies associated to its predecessors are not satisfied.	X

z/OS job status mapping

Table 22 on page 203 describes how a z/OS job status is mapped to the corresponding job internal status.

Table 22. z/OS job status mapping

This job status ...	Maps to this job internal status
Waiting	W
Ready	A, R, *
Running	S
Successful	C
Error	E
Canceled	I, D
Held	A,R,* manually held
Undefined	U
Suppressed by Condition	X

Related concepts:

“Job” on page 38

Related tasks:

“Creating a task to Monitor Jobs” on page 132

“Creating a task to Monitor Critical Jobs” on page 134

“Creating a task to Monitor Tasks on Multiple Engines” on page 137

Status description and mapping for distributed job streams

Distributed

There are the following types of status for job streams:

“Job stream status”

A subset of internal statuses that is common for both Tivoli Workload Scheduler for distributed and Tivoli Workload Scheduler for z/OS environments.

“Job stream internal status” on page 204

The Tivoli Workload Scheduler job stream status registered on the workstation where the job stream is running. The internal status uniquely identifies a job stream status in Tivoli Workload Scheduler.

Job stream status

Table 23 lists the job stream statuses.

Table 23. Job stream status

This job stream status ...	Means that ...
Waiting	The job stream is waiting for its dependencies to be resolved.
Ready	The dependencies of the job stream have been resolved and the job stream is ready to run.
Running	The job stream is running.
Successful	The job stream completed successfully.
Error	The job stream has stopped running with an error.
Canceled	The job stream was canceled.
Held	The job stream was interrupted.

Table 23. Job stream status (continued)

This job stream status ...	Means that ...
Undefined	The job stream status is currently being checked.
Blocked	The job stream was blocked because of unfulfilled dependencies.

Job stream internal status

Table 24 lists the job stream internal statuses.

Table 24. Job stream internal status

This job stream status ...	Means that ...
ABEND	The job stream terminated with a nonzero exit code.
ADD	The job stream was added with operator intervention.
CANCEL	The job stream was canceled.
CANCEL P	The job stream is pending cancellation. Cancellation is deferred until all of the dependencies, including an at time, are resolved.
EXEC	The job stream is running.
EXTRN	The job stream is in a remote Tivoli Workload Scheduler network and its status is unknown. An error occurred, a Rerun action was performed on the EXTERNAL job stream, or the INET job or job stream does not exist.
HOLD	The job stream is awaiting dependency resolution.
READY	The dependencies for the job stream have been met but the time restrictions for the job stream have not.
STUCK	The job stream was interrupted. No jobs are launched without operator intervention.
SUCC	The job stream completed with an exit code of zero.
Get Job Status Error	This is for internetwork job streams and specifies that an error occurred while checking for the remote status.

Job stream status mapping

Table 25 describes how a job stream status is mapped to the corresponding job stream internal status.

Table 25. Job stream status mapping

This job stream status ...	Maps to this job stream internal status
Waiting	ADD, PEND, WAIT, WAITD, INTRO, HOLD
Ready	READY
Running	EXEC
Successful	SUCC
Error	ABEND, FAILED
Canceled	CANCEL, HOLD, CANCEL P
Held	HOLD
Undefined	ERROR, EXTRN

Table 25. Job stream status mapping (continued)

This job stream status ...	Maps to this job stream internal status
Blocked	STUCK

Related concepts:

“Job stream” on page 38

Related tasks:

“Creating a task to Monitor Job Streams” on page 138

“Creating a task to Monitor Job Streams on Multiple Engines” on page 140

“Creating job stream definitions” on page 83

Status description and mapping for z/OS job streams

z/OS

There are the following types of status for z/OS job streams:

“z/OS job stream statuses”

A subset of internal statuses that is common for both Tivoli Workload Scheduler for distributed and Tivoli Workload Scheduler for z/OS environments.

“z/OS job stream internal statuses”

The Tivoli Workload Scheduler job stream statuses registered on the controller. The internal status uniquely identifies the status of a z/OS job stream in Tivoli Workload Scheduler.

z/OS job stream statuses

Table 26 shows the z/OS job stream statuses that are displayed by the Dynamic Workload Console.

Table 26. z/OS job stream status

This job stream status ...	Means that ...
Waiting	No job in the job stream has started.
Running	The job stream is running.
Successful	The job stream completed successfully.
Error	The job stream has stopped running with an error.
Canceled	The job stream was canceled.

z/OS job stream internal statuses

Table 27 shows the z/OS job stream internal statuses that are registered on the Dynamic Workload Console controller.

Table 27. z/OS job stream internal status

This job stream status ...	Means that ...
Waiting	No job in the job stream has started.
Started	The job stream is running.

Table 27. z/OS job stream internal status (continued)

This job stream status ...	Means that ...
Completed	The job stream completed successfully.
Error	The job stream has stopped running with an error.
Deleted	The job stream was deleted.
Undefined	The job stream status is not known.

z/OS job stream status mapping

Table 28 describes how a z/OS job stream status is mapped to the corresponding job stream internal status.

Table 28. z/OS job stream status mapping

This job stream status ...	Maps to this job stream internal status
Waiting	Waiting
Running	Started
Successful	Completed
Error	Error
Canceled	Deleted

Related concepts:

“Job stream” on page 38

Related tasks:

“Creating a task to Monitor Job Streams” on page 138

“Creating a task to Monitor Job Streams on Multiple Engines” on page 140

“Creating job stream definitions” on page 83

Workstation types

Table 29. Attribute settings for management workstation types

Attributes	Master domain manager	Domain manager	Backup domain manager
cpuname	The name of the workstation.		
description	A description for the workstation enclosed within double quotes. This attribute is optional.		
variable	The name of a variable table associated with the workstation. Variables used with the workstation are defined in this table. This attribute is optional.		
os	The operating system installed on the system. Specify one of the following values: UNIX WNT OTHER IBM_i		
node	The system host name or IP address.		
tcpaddr	The value assigned to <i>nm port</i> in the <i>localopts</i> file. For multiple workstations on a system, enter an unused port number. The default value is 31111.		
secureaddr	The value assigned to <i>nm ssl port</i> in the <i>localopts</i> file. Specify it if securitylevel is set to on, force or enabled.		

Table 29. Attribute settings for management workstation types (continued)

Attributes	Master domain manager	Domain manager	Backup domain manager
timezone tz	The time zone in which the system is located. It is recommended that the value matches the value set on the operating system.		
domain	MASTERDM	The name of the managed domain.	
host	Not applicable		
access	Not applicable		
type	manager		fta
ignore	Use this attribute if you do not want this workstation to appear in the next production plan.		
autolink	It indicates if a link between workstations is automatically opened at startup. Specify one of the following values: ON OFF This is an optional attribute. The default value is ON.		
behindfirewall	This setting is ignored.	It indicates if there is a firewall between the workstation and the master domain manager. Specify one of the following values: ON OFF The default value is OFF.	
securitylevel	The type of SSL authentication to use: enabled on force		
fullstatus	ON		
server	Not applicable		This setting is ignored.
protocol	Not applicable		
members	Not applicable		
requirements	Not applicable		

Table 30 describes the values you set for each attribute for target workstation types. Following the table you find additional details about each attribute.

Table 30. Attribute settings for target workstation types

Attribute	Fault-tolerant agent and standard agent	Workload broker workstation	Extended agent	Agent	Remote engine workstation	Pool	Dynamic pool
cpuname	The name of the workstation.						
description	A description for the workstation enclosed within double quotes. This attribute is optional.						
variable	The name of a variable table associated with the workstation. Variables used with the workstation are defined in this table. This attribute is optional.						

Table 30. Attribute settings for target workstation types (continued)

Attribute	Fault-tolerant agent and standard agent	Workload broker workstation	Extended agent	Agent	Remote engine workstation	Pool	Dynamic pool
os	The operating system installed on the system. Specify one of the following values: UNIX WNT OTHER IBM_i Specify OTHER for IBM i systems running as limited fault-tolerant agents.	OTHER	The operating system installed on the machine. Specify one of the following values: UNIX WNT OTHER IBM_i	This value setting is discovered on the system.	The operating system installed on the machine. Specify one of the following values: UNIX WNT ZOS	The operating system installed on the machine. Specify one of the following values: UNIX WNT OTHER IBM_i	
node	The system host name or IP address.		The system host name or IP address. Specify NULL when host is set to \$MASTER, or when defining an extended agent for PeopleSoft, SAP or Oracle.	Agent host name or IP address.	Remote engine host name or IP address.	Not applicable	
tcpaddr	The value assigned to <i>nm port</i> in the <i>localopts</i> file. When defining multiple workstations on a system, enter an unused port number. The default value is 31111.	The value assigned to <i>nm port</i> in the <i>localopts</i> file. When defining multiple workstations on a system, enter an unused port number. The default value is 41114.	See the selected access method specifications.	The port number to communicate with the agent when the protocol is http.	The port number to communicate with the remote engine when the protocol is http.	Not applicable	
secureaddr	The value assigned to <i>nm ssl port</i> in the <i>localopts</i> file. Specify it if securitylevel is set to on, force or enabled.	Not Applicable	Not Applicable	The port number to communicate with the agent when the protocol is https.	The port number to communicate with the remote engine when the protocol is https.	Not applicable	
timezone tz	The time zone in which the system is located. It is recommended that the value matches the value set on the operating system.		The time zone set on the workstation specified in the host attribute.	The time zone set on the agent.	The time zone set on the remote engine.	The time zone set on the pool agents.	The time zone set on the dynamic pool agents.
domain	Specify an existing domain. The default value for fault-tolerant agents is MASTERDM. This setting is mandatory for standard agents.	Specify an existing domain. This setting is mandatory.	This setting is needed only if the value assigned to host is \$MANAGER.	Not applicable			

Table 30. Attribute settings for target workstation types (continued)

Attribute	Fault-tolerant agent and standard agent	Workload broker workstation	Extended agent	Agent	Remote engine workstation	Pool	Dynamic pool
host	Not Applicable		The host workstation. It can be set to \$MASTER or \$MANAGER.	The broker workstation.			
access	Not Applicable			Select the appropriate access method file name.	Not Applicable		
agentID				The unique identifier of the dynamic agent			
type	fta s-agent The default value is fta. Specify fta for IBM i systems running as limited fault-tolerant agents.	broker	x-agent	agent	rem-eng	pool	d-pool
ignore	Use this attribute if you do not want this workstation to appear in the next production plan.						
autolink	It indicates if a link between workstations is automatically opened at startup. Specify one of the following values: ON OFF This is an optional attribute. The default value is ON.		OFF	Not applicable			
behindfirewall	It indicates if there is a firewall between the workstation and the master domain manager. Specify one of the following values: ON OFF The default value is OFF.		OFF	Not applicable			
securitylevel	The type of SSL authentication to use: enabled on force Not applicable for IBM i systems running as limited fault-tolerant agents.	Not Applicable					

Table 30. Attribute settings for target workstation types (continued)

Attribute	Fault-tolerant agent and standard agent	Workload broker workstation	Extended agent	Agent	Remote engine workstation	Pool	Dynamic pool
fullstatus	It indicates if the workstation is updated about job processing status in its domain and subdomains. Specify one of the following values: ON OFF Specify OFF for standard agents.	OFF		Not applicable			
server	0-9, A-Z. When specified, it requires the creation of a dedicated mailman processes on the parent workstation.		Not Applicable				
protocol	Not applicable			Specify one of the following values: http https This attribute is optional. When not specified, it is automatically determined from the settings specified for tcpaddr and secureaddr .		Not applicable	
members	Not applicable					Required value	Not applicable
requirements	Not applicable						Required value

Related concepts:

“Workstation” on page 31

Related tasks:

“Creating distributed workstations” on page 75

“Creating z/OS workstations” on page 76

“Creating z/OS virtual workstations” on page 77

“Creating a task to Monitor Workstations” on page 127


Graphical Views in the plan

You can use these views to monitor the progress of your job streams in the plan in a graphical map.

You can also take several actions on the objects displayed in the maps. Almost all the actions and information available in the traditional views resulting from Monitor tasks are also graphically available from these views.

All the panels provide a toolbar that you can use to act upon the views and the objects displayed. For more information about the actions available from the toolbars, see the specific panel help.

The actions that have any effect on the plan require that you click the **Refresh** button before their outcome is displayed. The action is submitted to the engine and the outcome becomes visible only after the engine has processed it.

By clicking the export icon , you can also export the displayed graphic in a Scalable Vector Graphics (SVG) file. With this type of file, vector-based images can be scaled indefinitely without degrading.

The following graphical views are currently available in the plan:

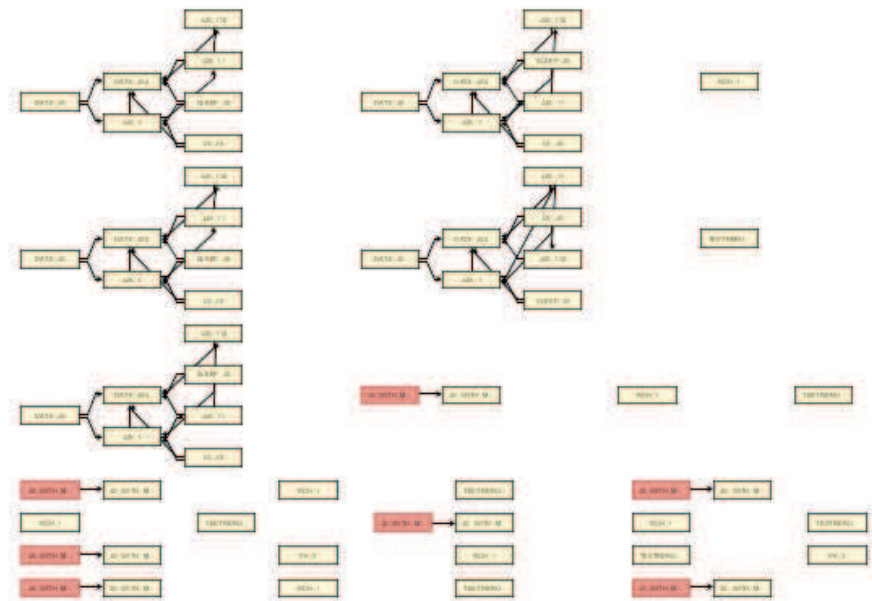
- “Plan View” (available only with Tivoli Workload Scheduler 8.5.1 or higher and Tivoli Workload Scheduler for z/OS 8.5.1 via z/OS Connector 8.5.1 or higher.)
- “Impact View” on page 212
- “Job Stream View” on page 212

Plan View

Use this view to get an overall picture of your plan.

This view shows a collapsed plan; it shows only the job streams, hiding any jobs and internal dependencies. External dependencies, both at job and job stream levels, are shown as arrows between job streams. If a job stream has multiple dependencies on another job stream, these are shown as one single arrow.

The following is an example of the view:



Because potentially a plan can contain a huge number of job streams, by default the Plan View displays a limited number of objects. You can determine which job streams to display by specifying filtering criteria in the **Show Plan View** task.

The maximum number of job streams displayed by the Plan View is 1000.

You can launch this view, in the following way:

1. From the portfolio, click: **Administration > Monitor > Show Plan View**.

2. From the list of Show Plan View tasks, click the task you want to run.

From the Plan View, you can modify the filters to show different job streams by clicking the icon, changing filter criteria, and running the task again. Consider that changes in the filters apply only to the current view; not to the task definition.

Impact View

This view has similar appearance and available actions to the Job Stream View.

An important difference is that the Impact View allows you to navigate through the displayed jobs and job streams, choosing the level of predecessors and successors to display.

You can open this view, in the following ways:

- From the table of results of a Monitor Jobs or Monitor Job Streams task, select an item and select **Graphical Views > Impact View**.
- From a Job Stream View, right-click a job or job stream and select **Open > Impact View**. In the new panel the object from which you started the Impact view is centered and highlighted.
- From an Impact View, right-click a job or job stream and select **Show > Impact View**. The same panel is refreshed, displaying a new view where the object from which you started the Impact view is centered and highlighted.
- From a Plan View, right-click a job stream and select **Open > Impact View**. In the new panel, the object from which you started the Impact view is centered and highlighted.

By default the Impact View displays all the internal predecessors and successors of the selected object and the first level of external predecessors and successors. Jobs that belong to the same job stream but that are not predecessors or successors of the selected job are not displayed, unless you choose a deeper dependency level, or switch to the job stream view.

The maximum number of dependencies displayed by the Impact View is 1000.

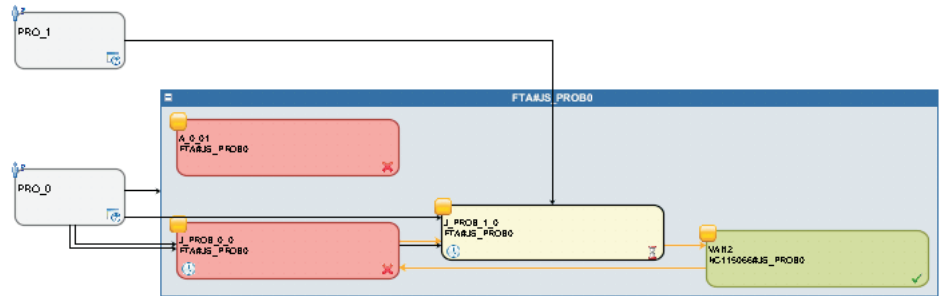
The actions you can take on the objects in the view are almost the same as those available in the Job Stream View, except for those regarding files, prompts, and resources, which are not displayed in the Impact View.

Job Stream View

This view displays a picture of the job stream with all its jobs and related dependencies. You can open this view in the following ways:

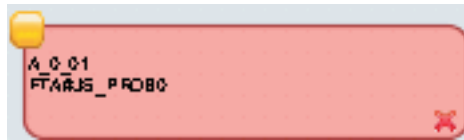
- From the table of results of a Monitor Jobs or Monitor Job Streams task, select an item and select **Graphical Views > Job Stream View**.
- From a Job Stream View or Impact View, right-click a job or job stream and select **Open > Job Stream View**.
- From the Plan View, right-click a job stream and select **Open > Job Stream View**.

The following is an example of Job Stream View:



The main elements in the Job Stream View are:

- **Jobs**



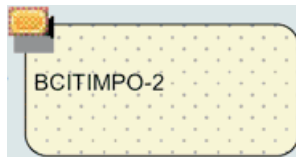
Their status is indicated by their colors and small icons at the bottom right corner. Further icons can be displayed inside the box to indicate:



The job is part of a rerun chain. By opening the job properties you can view further details about the rerun jobs, such as the outcome and job log of each rerun job.



The job or job stream is time-dependent.

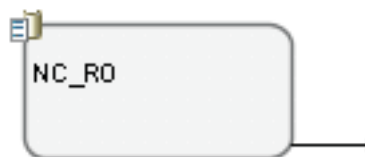


This image represents shadow jobs, which are jobs running locally that are used to map jobs running on the remote engine.

Further details are available in the tooltip displayed by hovering on the job with the cursor.

You can take multiple actions on the job by right-clicking it and choosing options from the context menu. The actions available depend on the selected engine and on the type of job. From this menu, you can view and print the job log and the job properties, or act on the job and its dependencies. You can also take actions on the job workstation or open the job and job stream definitions in the database to modify them.

- **Dependencies**



They are shown as smaller boxes, connected to the depending objects by arrows that represent the dependency relationships. To take actions on dependencies,

you can right-click either the box or the arrow and choose an option from the context menu. Icons displayed next to the box corners describe the dependency type and status. Further details are available in the tooltip displayed by hovering on the dependency with the cursor. The maximum number of dependencies displayed by the Job Stream View is 1000.

Related concepts:

“Plans” on page 60

“Plan view tasks” on page 132

Related tasks:

“Creating a task to Show Plan View” on page 157

Workload Designer

This section provides some reference information about the Workload Designer.

Working List

Use this pane to view all the objects you are working on, to create new objects, and to search for existing ones in the database.

The objects you see in this list are those you have currently open and you can switch between them to act on them in the workspace on the right.

In this pane you can use the following toolbars:



Use the toolbar to work with the objects displayed in this pane.



Use the toolbar to select or clear all the objects displayed in the list or to collapse and expand the list.

The object highlighted in the list is open in the workspace on the right, where you can edit it.

The object open in the workspace on the right is displayed in bold in this list.

The objects in the list can be marked by the following icons: The objects in the list can be marked by the following icons:



The object is open in edit mode.



The object has pending changes that must still be saved. Click the Save icon to commit the changes to the database.



The object is incorrect or incomplete and cannot be saved until all errors are fixed.



The object is currently being loaded. It cannot be edited until the loading completes.

If you close an object, it is removed from the working list. If it was locked, by closing it you also unlock it.

Related tasks:

“Designing your Workload” on page 79

“Editing objects from the Working List View” on page 80

Quick Open

Use this pane to rapidly open objects from the database.

To create dependencies as well as to add jobs to a job stream, you can drag one or more of the objects contained in the **Quick Open** pane and drop them in the Details pane or Graphical View in the workspace on the right.

Each of the following icons represents an object category. Click the icon corresponding to the object type that you want to open. **Distributed**



Job Definition



Remote SAP Standard R/3 Job



Job Stream



Prompt



Resource



User



Calendar



Workstation Class



Variable Table



Workload Application Template

z/OS



Job Stream



Resource



Period



Variable Table



Operator Instructions



Event Tracking Criterion



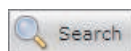
Workstation Closed Interval



z/OS job



Remote SAP Standard R/3 Job

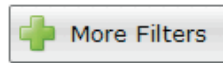


Click this button to start the search, after having optionally specified your

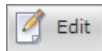
search criteria. If you leave a field blank or specify a single asterisk (*), no specific search criterion is used. The available fields vary depending on the object type you select.

Wildcard characters are supported and used as follows: the asterisk (*) indicates multiple characters and the question mark (?) indicates one single character.

All the objects resulting from your search are shown up to a maximum of 250 items. If the list is longer, use filters to refine your search and reduce the number of results:



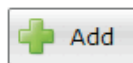
Hover with the cursor over an object to see its description displayed in a tooltip.



Click this button to open the selected objects in edit mode. The objects are locked in the database until you finish editing and save them. The objects you open are listed in the **Working List** pane.



Click this button to open the selected objects in read-only mode.



If you are editing a job stream, you can search for jobs or dependencies and click this button to add them to the selected object. You can also drag the objects contained in the **Quick Open** pane and drop them in the Details pane or **Graphical View** to create dependencies and to add jobs to the job stream.

Related tasks:

"Designing your Workload" on page 79

"Editing objects from the Quick Open View" on page 81

Details view

Use this view to create, display, and modify job streams, and to add jobs, run cycles, and dependencies to them. It shows the job stream with all the jobs it contains, associated dependencies, and run cycles.

This view displays the structure of the objects in a tree table. It applies to

Distributed job streams

z/OS job streams and resources.

z/OS It also shows the resource availability intervals. Use the view also to create, display, and modify resources, and to add availability intervals to them.

You can drag objects from the **Quick Open** section and drop them in these panes to:

- Add jobs to the job stream.
- Add external dependencies to the job stream and its jobs.

For example, you can:

- Drag a job definition and drop it into a job stream to add a new job to the job stream.
- Drag a job definition and drop it outside a job stream to create a job stream external dependency.
- Drag a job definition and drop it over a job in the job stream to create a job external dependency.
- Drag a resource, prompt, or job stream and drop it over a job or job stream to create an external dependency.

In this view you can use the following toolbar, icons, and buttons:



All the actions available using the icons of the toolbar act on the root object of the tree table.



Use this button to print the contents of the tree in text format for job streams. Click the button to open a new page containing the information in the tree table for the job stream you are viewing. The output is organized in columns. For all job streams, the object properties shown are Name, Type, and Workstation.

z/OS

For z/OS job streams, also Task Type and Owner are shown.



This icon is located at the end of each row and in the objects pop-up menu. Click it to remove the corresponding object from the tree table. Use it, for example, to remove a job from a job stream or a dependency from a job. The removal becomes effective only when you save the object.

Select an Action ▾

Use this button to act on objects located at inner levels. Select the object in the tree table and open the menu, or use the pop-up menu by right-clicking the object. This button is enabled only if the operation is available for the selected object.

Some of the actions available from the action menus are:

Add Objects Selected in Working List

Add all the objects currently selected in the **Working List**. The objects are added as part of the flow or as dependencies. For example, adding a job definition to a job stream adds it to the job stream flow, while adding a prompt adds it as a dependency.

Add object

Add any specific object listed in the option.

Open Job Definition

Open the definition of the job to display its properties as they are defined in the database.

Copy

Copy the object and keep it in memory until you either copy another object or close the Workload Designer window. You can only copy a job stream if it already exists in the database (that is, it must have been saved at least once). You can also copy dependencies, except for internal job dependencies.

Paste as Dependency

Paste the object previously copied as a dependency of the selected one. The copied object can be a job, a job stream, or any kind of dependency. You can only paste source objects that already exist in the database.

You can paste a job stream as:

- External job stream dependency of the same or of another job stream
- External job stream dependency of a job in the same or in another job stream

You can paste a job as:

- External job dependency of itself or of the job stream it belongs to
- Internal job dependency of another job in the same job stream
- External job dependency of another job stream or job in another job stream

The object selected in the tree table is highlighted in light blue and is displayed in bold in the **Working List**. When you select an object in the tree table, its properties are displayed in the lower pane, where you can view and edit them, if you are authorized to do so.

Related tasks:

“Designing your Workload” on page 79

“Editing objects from the Details View” on page 82

Graphical View

This view shows the job stream with all the jobs it contains and their associated dependencies. When you select an object in the graphical view, the object properties are displayed at the bottom of the properties pane.

The same information and actions available from the **Details** pane are also available from the **Graphical View**.

Note: If you are using Internet Explorer version 8 or 9, to correctly display the Graphical View, you must add the hostname of the Dynamic Workload Console to the web sites used in Compatibility View by Internet Explorer. To do it, from Internet Explorer toolbar, click **Tools > Compatibility View Settings** and add the Dynamic Workload Console hostname to the list.

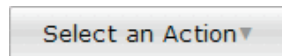
You can drag objects from the **Quick Open** section and drop them in these panes to:

- Add jobs to the job stream.
- Add external dependencies to the job stream and its jobs.

For example, you can:

- Drag a job definition and drop it into a job stream to add a new job to the job stream.
- Drag a job definition and drop it outside a job stream to create a job stream external dependency.
- Drag a job definition and drop it over a job in the job stream to create a job external dependency.
- Drag a resource, prompt, or job stream and drop it over a job or job stream to create an external dependency.

In this view you can use the following toolbar, icons, and buttons:



Use this button to perform multiple actions. Select the object in the graphic and open the menu, or use the pop-up menu by right-clicking the object. This button is enabled only if this operation is available for the selected object.



Use the toolbar to rapidly create dependency relationships, remove objects, or move and optimize the view.



Use this icon to resize the graph to its best view.



Click this icon and select an object to highlight all the incoming and outgoing dependencies.



Use this icon to create dependencies. Click the icon, click an object, and then draw a line to the job stream or to the job that you want to depend on the object. You can only use this icon to create dependencies from objects displayed in the view and by drawing lines in the correct direction (from the dependency to the depending object).

z/OS You can use this icon also to create a conditional dependency on jobs internal to the job stream. However, in this case, the job stream cannot be saved until you have manually updated the conditional dependency table, by specifying all the required information about the condition.



Click this icon to remove the selected object from the view. Use it, for example, to remove a job from a job stream or a dependency from a job. The removal becomes effective only when you save the object.



Click this icon to refresh the view.



Click this icon to launch short videos that provide some tips about what you can do from this panel. From the displayed list, click the demo you want to watch. These and more demos are stored on the Tivoli Workload Scheduler Wiki Media Gallery and are available in English only.

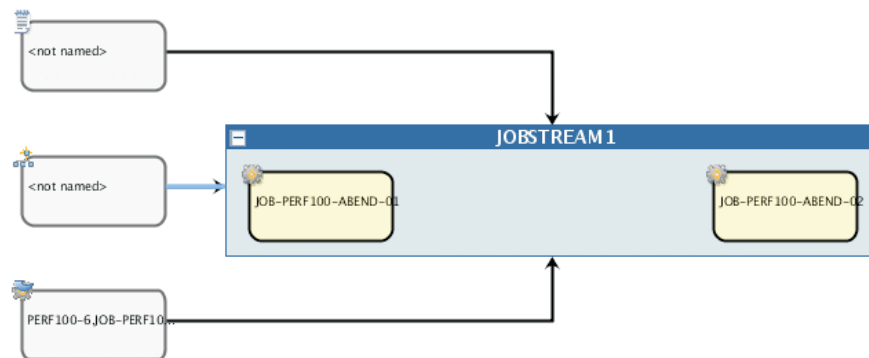


Click this icon to export the graphic in a Scalable Vector Graphics (SVG) file. With this type of file, vector-based images can be scaled indefinitely without degrading. A vector graphics program uses these mathematical formulas to construct the screen image, building the best quality image possible, given the screen resolution.

Hover with the cursor over an object to display a tooltip containing more object information.

The object selected in this view is displayed in bold in the **Working List** and its properties are shown in the pane at the bottom of the view.

The following picture is an example of a graphical representation of a job stream.

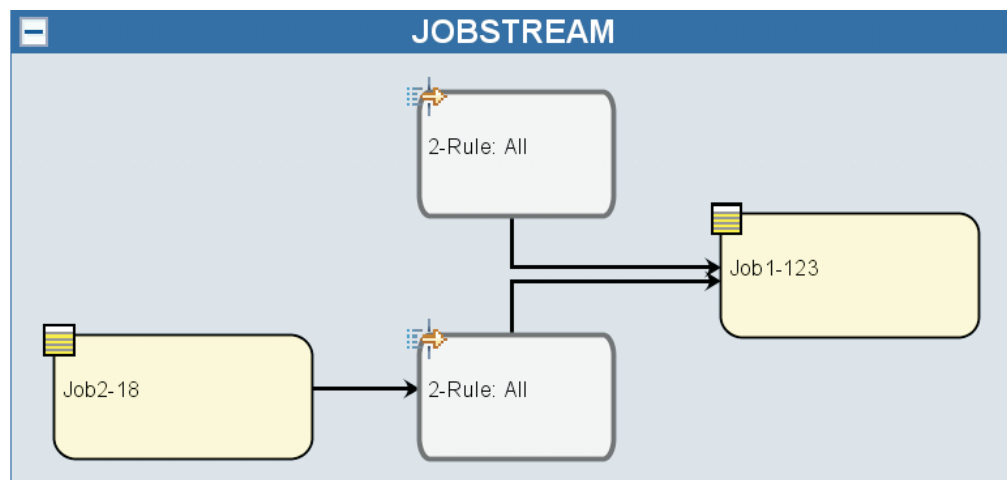


Dependencies

When you click a job stream or job dependency, you select its dependency relationship (which is highlighted in light blue) and you can copy or remove it. If this object is a dependency for multiple items, click it again to select the next dependency relationship.

The arrows represent the dependency relationships; where the arrow goes from the dependency to the object that depends on it.

z/OS Dependencies can also be conditional. In the graphical view, this type of dependency is represented by a box with an icon, which is the target of all the arrows spawned from the objects defined as conditions to be satisfied. The box is also the starting point for the arrow targeted to the depending objects, as shown in the following picture.



Note: The graphical view is powered by IBM ILOG JViews. For more information, see <http://www.ilog.com>.

Related tasks:

“Designing your Workload” on page 79

“Editing objects from the Graphical View” on page 82

Scalable Vector Graphics

Scalable Vector Graphics (SVG) is a file format for describing two-dimensional vector graphics.

It can natively represent geometrical primitives such as lines, curves, and shapes, and is therefore an ideal format for storing diagrams and graphs. Because Scalable Vector Graphics files contain geometrical objects, images can be freely transformed or zoomed without decreasing their quality, and usually print very crisply even when resized.

Scalable Vector Graphics format is supported by many graphics applications as well as by many browsers.

Object properties

You can use the Properties pane to modify the properties of the selected object.

The properties pane is divided into tabs, which contain options and properties relating to the object currently open.

If you have more than one object open in the **Working List**, the properties displayed in this pane are relating to the object currently selected in the **Details** or **Graphical** view above.

Note: For all the details about options and fields displayed in the panels, see the online help by clicking the question mark located at the top-right corner of each panel.

Run Cycle Preview

This view applies only to job streams and shows the run cycles defined for the selected job stream.

It shows the days on which the job stream runs and the days on which it is explicitly excluded from running. You can see this view as a Month or full Year view, and use the navigation toolbar to go to the next or previous Month or Year.

When you modify a run cycle, the view is automatically updated.

Color coding and symbols are used to indicate included and excluded days in the run cycle rule.

Selecting a run cycle, the colors have the following meanings:



The run cycle includes that day.



The run cycle excludes that day.

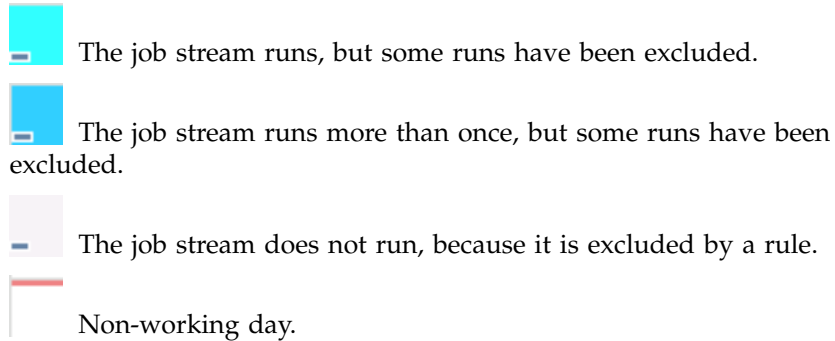
Selecting a job stream, the colors have the following meanings:



The job stream runs only once on that day.



The job stream runs more than once on that day.



Click any highlighted day to view details about the run cycles occurring on the selected day.

Related concepts:

“Run cycle” on page 41

“Run cycle group” on page 43

“Creating and managing run cycle groups and their run cycles” on page 100

Related tasks:

“Designing your Workload” on page 79

Message History

Use this panel to view all the messages related to the active session.

When you open this section, by default all warning and error messages of the current session are displayed.

Click the error, warning or information icon to filter for the corresponding messages. No message is displayed until you click an icon.



Click this icon to view error messages.



Click this icon to view warning messages.



Click this icon to view informational messages.



Click this icon to clear all messages from the list, regardless of whether they are displayed or filtered out.



Click this icon to clear only the messages currently displayed.

Find Type the string you want to find in the messages. Typing in the text fields automatically triggers the search, you do not have to click any further button. To reset the search criteria, clear the **Find** text field.

Return

Click this button to go back to the previous view, hiding the Message History view.

Related tasks:

“Designing your Workload” on page 79

Using recovery options in job definitions

The recovery options indicate the actions to be taken if a job fails.

The following table summarizes possible combinations of recovery options and actions. The table is based on the following criteria from a job stream called **sked1**:

- Job stream **sked1** has two jobs, **job1** and **job2**.

- If selected for **job1**, the recovery job is **jobr**.
- **Job2** is dependent on **job1** and will not start until **job1** is complete.

Table 31. Recovery option table

Prompt/Job	Stop	Continue	Rerun
Recovery prompt: No Recovery job: No	Intervention is required.	Run job2 regardless of job1 completion status.	<ul style="list-style-type: none"> • Rerun job1. <ul style="list-style-type: none"> – If job1 ends in error, issue scheduler prompt. – If reply is yes, repeat the above steps. • If job1 is successful, run job2.
Recovery prompt: Yes Recovery job: No	Issue recovery prompt. Intervention is required.	<ul style="list-style-type: none"> • Issue recovery prompt. • If reply is yes, run job2 regardless of job1 completion status. 	<ul style="list-style-type: none"> • Issue recovery prompt. <ul style="list-style-type: none"> – If reply is yes, rerun job1. – If job1 ends in error, repeat the above steps. • If job1 is successful, run job2.
Recovery prompt: No Recovery job: Yes	<ul style="list-style-type: none"> • Run jobr. <ul style="list-style-type: none"> – If jobr ends in error, intervention is required. – If jobr is successful, run job2. 	<ul style="list-style-type: none"> • Run jobr. • Run job2 regardless of job1 completion status. 	<ul style="list-style-type: none"> • Run jobr. <ul style="list-style-type: none"> – If jobr ends in error, intervention is required. – If jobr is successful, rerun job1. • If job1 ends in error, issue scheduler prompt. <ul style="list-style-type: none"> – If reply is yes, repeat the above steps. – If job1 is successful, run job2.

Table 31. Recovery option table (continued)

Prompt/Job	Stop	Continue	Rerun
Recovery prompt: Yes Recovery job: Yes	<ul style="list-style-type: none"> • Issue recovery prompt. <ul style="list-style-type: none"> – If reply is yes, run jobr. - If jobr ends in error, intervention is required. - If jobr is successful, run job2. 	<ul style="list-style-type: none"> • Issue recovery prompt. <ul style="list-style-type: none"> – If reply is yes, run jobr. • Run job2 regardless of job1 completion status. 	<ul style="list-style-type: none"> • Issue recovery prompt. <ul style="list-style-type: none"> – If reply is yes, run jobr. - If jobr ends in error, intervention is required. - If jobr is successful, rerun job1. • If job1 ends in error, repeat the above steps. • If job1 is successful, run job2.

Note:

- *Intervention is required* means that job2 is not released from its dependency on job1, and therefore must be released by the operator. You can also manually rerun job1 or cancel it.
- The **continue** recovery option overrides the abend state, which might cause the schedule containing the job ended in error to be marked as successful. This prevents the schedule from being carried forward to the next day.
- If you select the **Rerun** option without supplying a recovery prompt, when the job is unsuccessful Tivoli Workload Scheduler creates a prompt that asks if you want to proceed.
- To reference a recovery job in **conman**, you must use the name of the original job (job1 in the scenario above, not jobr). Recovery jobs are run only one per abend.

Not all jobs are eligible to have recovery jobs run on a different workstation. Follow these guidelines:

- If either workstation is an extended agent, it must be hosted by a domain manager or a fault-tolerant agent that runs in **Full Status** mode.
- The recovery job workstation must be in the same domain as the parent job workstation.
- If the recovery job workstation is a fault-tolerant agent, it must run in **Full Status** mode.

Reports

This section contains examples of regular expressions and SQL reports. Some samples of report output are also shown.

Related concepts:

“Reports” on page 66

Related reference:

Chapter 14, “Reporting,” on page 165

Regular Expressions

This section gives some examples of useful regular expressions, together with a table that defines the expressions supported by Tivoli Workload Scheduler. Further information about regular expressions is also widely available in the Internet.

Useful regular expressions

The following table shows some useful regular expressions for use with the plan extractor, both for filtering jobs and job streams, and for configuring business unit names.

Table 32. Useful regular expressions

Requirement	Regular expression	Example
To obtain the same effect as using the "@"character in the Tivoli Workload Scheduler command line	.*	-JScpu .*
		Used as a parameter to the plan extractor, filters for all job stream workstations.
To join different criteria in an "OR" relationship		(TIV.*) (.IBM.*)
		Filters for all items that begin with the string "TIV" or that contain the string "IBM" (regular expressions are case-sensitive).
To select objects that begin with one of several characters	[<the characters to be included>]	[ABC].*
		Filters for all items that begin with either "A", "B", or "C".
To select objects that do not begin with one of several characters	[^<the characters to be excluded>]	[^ABC].*
		Filters for all items that do not begin with either "A", "B", or "C".
To select objects in which certain characters appear a certain number of times	<the character to be counted>{<the character count>}	A{3}.*
		Filters for all items that begin with the string "AAA".
To select objects in which certain characters appear at least a certain number of times	<the character to be counted>{<the character count>}	A{3,}.*
		Filters for all items that begin with the string "AAA", "AAAA", "AAAAA", and so on.
To select objects in which certain characters appear at least a certain number of times, but not more than a certain number of times	<the character to be counted>{<the lower character count>,<the upper character count>}	A{3,4}.*
		Filters for all items that begin with the string "AAA", or "AAAA"; a string that began with "AAAAA" would not be selected.

Complex expressions

These individual regular expressions can be combined to make a complex expression, as shown in the following table.

Table 33. Complex expressions

Example requirement	Regular expression
Select all strings that begin with "AA", "AB", "AC", "BA", "BB", "BC", "CA", "CB", or "CC", and also those that do not end in "X", "Y", or "Z".	<code>([ABC]{2}.*) (.^[^XYZ])</code>
Select all strings that begin with "AA" followed by either one or more numbers or one or more letters, and then by the character "_". It can finish with any characters. This would, for example, select the string AA11_XYZ76 and the string AAFGH_43KKK, but not the string AA8H_3232IHSDG, because this latter has both numbers and letters between the "AA" and the "_".	<code>A{2}([0-9]+ [A-Z]+)_.*</code>

Regular expressions supported by the plan extractor

The following tables provide full details of the regular expressions supported by the plan extractor.

Table 34. Regular expressions supported by the plan extractor: character

Syntax	Filters for?
<code>x</code>	The character <code>x</code> (regular expressions are case-sensitive)
<code>\\</code>	The backslash character
<code>\0n</code>	The character with octal value <code>0n</code> ($0 \leq n \leq 7$)
<code>\0nn</code>	The character with octal value <code>\0nn</code> ($0 \leq n \leq 7$)
<code>\0mnn</code>	The character with octal value <code>\0mnn</code> ($0 \leq m \leq 3, 0 \leq n \leq 7$)
<code>\0xhh</code>	The character with hexadecimal value <code>0xhh</code>
<code>\uhhhh</code>	The character with hexadecimal value <code>0xhhhh</code>
<code>\t</code>	The tab character (<code>'\u0009'</code>)
<code>\n</code>	The newline (line feed) character (<code>'\u000A'</code>)
<code>\r</code>	The carriage-return character (<code>'\u000D'</code>)
<code>\f</code>	The form-feed character (<code>'\u000C'</code>)
<code>\a</code>	The alert (bell) character (<code>'\u0007'</code>)
<code>\e</code>	The escape character (<code>'\u001B'</code>)
<code>\cx</code>	The control character corresponding to <code>x</code>

Table 35. Regular expressions supported by the plan extractor: character classes

Syntax	Filters for?
<code>[abc]</code>	<code>a</code> , <code>b</code> , or <code>c</code> (simple class)
<code>[^abc]</code>	Any character except <code>a</code> , <code>b</code> , or <code>c</code> (negation)
<code>[a-zA-Z]</code>	<code>a</code> through <code>z</code> or <code>A</code> through <code>Z</code> , inclusive (range)
<code>[a-d[m-p]]</code>	<code>a</code> through <code>d</code> , or <code>m</code> through <code>p</code> : <code>[a-dm-p]</code> (union)
<code>[a-z&&[def]]</code>	<code>d</code> , <code>e</code> , or <code>f</code> (intersection)
<code>[a-z&&[^bc]]</code>	<code>a</code> through <code>z</code> , except for <code>b</code> and <code>c</code> : <code>[ad-z]</code> (subtraction)
<code>[a-z&&[^m-p]]</code>	<code>a</code> through <code>z</code> , and not <code>m</code> through <code>p</code> : <code>[a-lq-z]</code> (subtraction)

Table 36. Regular expressions supported by the plan extractor: predefined character classes

Syntax	Filters for?
.	Any character (might or might not match line terminators)
\d	A digit: [0-9]
\D	A non-digit: [^0-9]
\s	A whitespace character: [\t\n\x0B\f\r]
\S	A non-whitespace character: [^\s]
\w	A word character: [a-zA-Z_0-9]
\W	A non-word character: [^\w]

Table 37. Regular expressions supported by the plan extractor: POSIX character classes (US-ASCII only)

Syntax	Filters for?
\p{Lower}	A lowercase alphabetic character: [a-z]
\p{Upper}	An uppercase alphabetic character: [A-Z]
\p{ASCII}	All ASCII: [\x00-\x7F]
\p{Alpha}	An alphabetic character: [\p{Lower}\p{Upper}]
\p{Digit}	A decimal digit: [0-9]
\p{Alnum}	An alphanumeric character: [\p{Alpha}\p{Digit}]
\p{Punct}	Punctuation: One of !"#\$%&'()*+,-./:;<=>?@[\]^_`{ }~
\p{Graph}	A visible character: [\p{Alnum}\p{Punct}]
\p{Print}	A printable character: [\p{Graph}]
\p{Blank}	A space or a tab: [\t]
\p{Cntrl}	A control character: [\x00-\x1F\x7F]
\p{XDigit}	A hexadecimal digit: [0-9a-fA-F]
\p{Space}	A whitespace character: [\t\n\x0B\f\r]

Table 38. Regular expressions supported by the plan extractor: classes for Unicode blocks and categories

Syntax	Filters for?
\p{InGreek}	A character in the Greek block (simple block)
\p{Lu}	An uppercase letter (simple category)
\\p{Sc}	A currency symbol
\P{InGreek}	Any character except one in the Greek block (negation)
[\p{L}&&[^\p{Lu}]]	Any letter except an uppercase letter (subtraction)

Table 39. Regular expressions supported by the plan extractor: boundary matchers

Syntax	Filters for?
^	The beginning of a line
\$	The end of a line
\b	A word boundary
\B	A non-word boundary
\A	The beginning of the input

Table 39. Regular expressions supported by the plan extractor: boundary matchers (continued)

Syntax	Filters for?
\G	The end of the previous match
\Z	The end of the input but for the final terminator, if any
\z	The end of the input

Table 40. Regular expressions supported by the plan extractor: greedy quantifiers

Syntax	Filters for?
X?	X, once or not at all
X*	X, zero or more times
X+	X, one or more times
X{n}	X, exactly <i>n</i> times
X{n,}	X, at least <i>n</i> times
X{n,m}	X, at least <i>n</i> but not more than <i>m</i> times

Table 41. Regular expressions supported by the plan extractor: reluctant quantifiers

Syntax	Filters for?
X??	X, once or not at all
X*?	X, zero or more times
X+?	X, one or more times
X{n}?	X, exactly <i>n</i> times
X{n,}?	X, at least <i>n</i> times
X{n,m}?	X, at least <i>n</i> but not more than <i>m</i> times

Table 42. Regular expressions supported by the plan extractor: possessive quantifiers

Syntax	Filters for?
X?+	X, once or not at all
X*+	X, zero or more times
X++	X, one or more times
X{n}+	X, exactly <i>n</i> times
X{n,}+	X, at least <i>n</i> times
X{n,m}+	X, at least <i>n</i> but not more than <i>m</i> times

Table 43. Regular expressions supported by the plan extractor: logical operators

Syntax	Filters for?
XY	X followed by Y
X Y	Either X or Y
(X)	X, as a capturing group

Table 44. Regular expressions supported by the plan extractor: back references

Syntax	Filters for?
\n	Whatever the <i>n</i> th capturing group matched

Table 45. Regular expressions supported by the plan extractor: quotation

Syntax	Filters for?
\	Nothing, but quotes the following character
\Q	Nothing, but quotes all characters until \E
\E	Nothing, but quotes all characters until \Q

Table 46. Regular expressions supported by the plan extractor: special constructs (non-capturing)

Syntax	Filters for?
(?:X)	X, as a non-capturing group
(?idsux-idsux)	Nothing, but turns match flags on - off
(?idsux-idsux:X)	X, as a non-capturing group with the given flags on - off
(?=X)	X, via zero-width positive lookahead
(?!X)	X, via zero-width negative lookahead
(?<=X)	X, via zero-width positive lookbehind
(?<!X)	X, via zero-width positive lookbehind
(?>X)	X, as an independent, non-capturing group

SQL report examples

This section provides some examples of queries that can be run using the SQL custom reports.

Jobs grouped by return codes

For each return code, this query returns the number of jobs that ended with the corresponding return code:

```
SELECT DISTINCT return_code AS RC count(job_name) AS ,#JOB
FROM mdl.job_history_v
GROUP BY return_code
```

Table 47. Example of query outcome

RC	#JOB
0	1670
5	11
6	1
50	2
127	352

Job statistics grouped on job status

For each job status, this query returns the number of jobs that ended with the corresponding job status and also the planned duration time, the total elapsed time, and total CPU time:

```

SELECT job_status, count(job_name) AS job count, floor(sum(planned_duration/
1000)) AS
planned duration, floor(sum(total_elapsed_time/1000)) AS total elapsed,
floor(sum(total_cpu_time/1000)) AS total cpu
FROM mdl.job_history_v GROUP BY job_status

FROM mdl.job_history_v

GROUP BY return_code

```

Table 48. Example of query outcome

JOB_STATUS	JOB COUNT	PLANNED DURATION	TOTAL ELAPSED	TOTAL CPU
A	366	0	21960	0
S	1670	1413360	1423500	183

Jobs in a range of return code

This query returns the number of job in a range of return codes

```

SELECT *

FROM (select DISTINCT return_code, count(job_name) AS #JOB

FROM mdl.job_history_v

GROUP BY return_code) AS temp

WHERE return_code > 0 AND return_code < 6

```

Table 49. Example of query outcome

RETURN_CODE	#JOB
5	11

Jobs that ran within a time range and finished with a specific job status

```

SELECT WORKSTATION_NAME, JOB_NAME, JOB_RUN_DATE_TIME

FROM MDL.JOB_HISTORY_V

WHERE JOB_RUN_DATE_TIME BETWEEN '2008-05-19 10:00:00.0' AND '2008-05-19
21:00:00.0' AND JOB_STATUS <> 'S'

ORDER BY JOB_RUN_DATE_TIME

```

Table 50. Example of query outcome

WORKSTATION_NAME	JOB_NAME	JOB_RUN_DATE_TIME
NC122072	PEAK_A_06	2008-08-03 23:23:00.0

Table 50. Example of query outcome (continued)

WORKSTATION_NAME	JOB_NAME	JOB_RUN_DATE_TIME
NC122072	JOB_RER_A	2008-08-03 23:23:00.0
NC122072	PEAK_A_13	2008-08-03 23:23:00.0
NC122072	PEAK_A_20	2008-08-03 23:23:00.0
NC122072	PEAK_A_27	2008-08-03 23:23:00.0
NC122072	PEAK_A_43	2008-08-03 23:23:00.0
NC122072	PEAK_B_19	2008-08-03 23:24:00.0

Related tasks:

“Creating a task to Create Custom SQL Reports” on page 170

Event rule

An *event rule* defines a set of actions that run when specific event conditions occur. An event rule definition correlates events and trigger actions.

Related concepts:

“Event management” on page 64

“Event management configuration” on page 9

Related tasks:


“Creating an event rule” on page 109

Action properties

When you select an action, its properties are displayed at the bottom of the panel, where you can edit them. Mandatory property values are displayed by default, marked by asterisks and with a yellow background. If you try to add an action without specifying one or more mandatory values, the action turns red and the rule cannot be saved.

You can add further properties by selecting them from the drop-down list. When you add new properties, they become mandatory, therefore you cannot leave them blank, but you can delete them by clicking the close icon.

You can also add the same property multiple times assigning different values to it. In this case all these properties are logically correlated by the conjunction *and*, creating a cumulative filter.

For examples, when you define your event filter, if you want to exclude some files from it, you can define as event properties all the filenames that *match* /tmp/tool/oldfiles, and all the filenames that *do not match* /tmp/tools/newfiles. Additionally, you can also remove the non-required properties, by clicking the close icon  next to them. You cannot remove mandatory properties.

Using variable information into action properties

To better qualify your action, you can use some event properties as variable information that can be added to the action properties.

For example, you can include the job name in the mail body, if you have mail notification as a response action to a job-related event. You can include this

variable information together with normal text in action properties that require a string value. For those properties that require a numeric value, you can enter either the variable information or a number.

The event properties that can be used as variable information for the actions can be selected from the list that is displayed if you click the button **Variable**. Select **Machine-readable format** check box when you want to use the variable as input to a command or a script. Alternatively, you can recall this information within the action property by copying the *event alias* (displayed in the first row of the event box) and pasting it into the action property field, complying with the following syntax:

`%{event_alias.property_name}`

Use it to include normal text information. This can be useful if you want to see this text in a message or an email.

`${event_alias.property_name}`

Use it to include machine-readable information. This can be useful if you want to use the action as input to a command or a script.

Event properties


When you select an event, its properties are displayed at the bottom of the panel, where you can edit them. A tooltip over the event box shows all the event properties available as variables that you can use to define the action properties.


When you choose the event properties, you define a *filter* for all the events that you want to monitor and manage. The most meaningful properties you choose are logically correlated and represent the *event scope*, which is displayed in the event box.


You can add further properties by selecting them from the drop-down list. When you add new properties, they become mandatory, therefore you cannot leave them blank but you can delete them by clicking the close icon.


Mandatory property values are displayed by default, marked by asterisks and with a yellow background. If you try to add an event without specifying one or more mandatory values, the event turns red, the invalid fields are highlighted, and the rule cannot be saved.

You can also add the same property multiple times assigning different values to it. In this case all these properties are logically correlated by the conjunction *and*, creating a cumulative filter.

For example, when you define your event filter, if you want to exclude some files from it, you can define as event properties all the filenames that *match* `/tmp/tool/oldfiles`, and all the filenames that *do not match* `/tmp/tools/newfiles`. You can also remove the non-required properties, by clicking the close icon  next to them. You cannot remove mandatory properties.

When completing this section, consider that wildcard characters are supported, depending on the event type. When wildcard characters are supported, the wildcard icon  is enabled.

Event properties that do not support wildcard characters are identified by the wildcards not supported icon .

Some event properties allow you to specify multiple values separated by semicolon (;). These properties are identified by the multiple filter icon . When you specify multiple values, these values are logically correlated by the conjunction *or*.

For example, if you create a **Job Status Changed** event specifying the **Job name** property as "A;B;C", an action is triggered each time either of the specified jobs changes its status.

Event properties that do not support multiple values are identified by this icon



Activation statuses of the event rules

The following list shows the possible statuses of the event rules:



Active

This status includes rules that are in any of the following internal conditions:

- Active
- Update Pending
- Error During Update
- Deactivation Pending
- Error During Deactivation



Inactive

This status includes rules that are in any of the following internal conditions:

- Inactive
- Activation Pending (an intermediate status that occurs, for example, when you change a rule, save these changes, and set the rule to **Non Draft**)
- Error During Activation

Schemas for generic events

This section lists some schemas that you can use to produce a customized event.

Note: For PDF users, the parameter tables listed below are files referenced by the PDF. They are not saved locally with the PDF from the infocenter. You must first view them on the infocenter before saving or printing.

- **Action definitions**
- **Event definition**
- **Common definitions**

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol ([®] or [™]), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.



Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Index

A

- abend prompt 50
- access method jobs 84, 91
- accessibility ix
- accessing
 - online publications 183
- action
 - generic 65
 - notification 65
 - operational 65
 - properties
 - using variable information 231
- activation status
 - event rules 233
- ad-hoc prompt 50
- adding
 - a job to a job stream 94
 - dependency 98
 - task
 - to favorite bookmarks 23
- agents pool
 - creating 79
- application monitoring event 64
- archived plan 61
- associating
 - plan
 - to engine 156
 - to task 155
- auditing
 - Self-Service Catalog 193
 - Self-Service Dashboards 193

B

- backup domain manager
 - attributes 75, 77
 - configuring 75, 77
- backup dynamic domain manager
 - attributes 75, 77
 - configuring 75, 77
- backup master domain manager 31
 - attributes 75, 77
 - configuring 75, 77
- beacon 24, 25, 26, 185
 - send internal communications 26
- bookmark
 - adding tasks to 23

C

- calendar
 - definition 41
 - holidays 41
- call to a Web service
 - sample JSDL files 84, 91
- changing
 - settings repository user 18
 - user password 121
- Cognos prompt type
 - date syntax 94

- Cognos prompt type (*continued*)
 - time stamp syntax 94
 - time syntax 94
- communication
 - type based on SSL communication
 - options 198
- complex expressions 225
- condition dependencies
 - definition 49, 51
 - examples 53
 - handling recovery 54
 - step level 54
- conditional logic 51
- conditional recovery job option 54
- configuring
 - settings repository 16, 19
- console
 - customizing 21
 - portfolio 1
 - start 1
- controlling
 - job processing
 - by using dependencies 149
 - by using job confirmation 152
 - by using job priority 151
 - by using job recovery actions 152
 - by using limits 151
 - by using time restrictions 150
 - by using workstation fence 151
 - job stream processing
 - by using dependencies 149
 - by using job priority 151
 - by using limits 151
 - by using time restrictions 150
 - by using workstation fence 151
- conventions used in publications ix
- creating
 - cross dependencies 99
 - database objects 79
 - distributed workstation 75
 - domain 78
 - Dynamic Workload Broker
 - objects 107
 - event rule 109
 - objects in the database 75
 - pool of agents 79
 - task
 - generate Custom SQL report 170
 - generate Job Run History report 166
 - generate Job Run Statistics report 165
 - generate Plan report 168
 - generate Workstation Workload Runtimes report 168
 - generate Workstation Workload Summary report 167
 - list event rule definitions 116
 - monitor critical jobs 134
 - monitor domains 129
 - monitor event rules 145

- creating (*continued*)
 - task (*continued*)
 - monitor files 141
 - monitor job streams 125, 138
 - monitor job streams on multiple engines 140
 - monitor jobs 125, 132
 - monitor jobs on multiple engines 137
 - monitor operator messages 148
 - monitor prompts 143
 - monitor resources 144
 - monitor triggered actions 147
 - monitor workstations 125, 127
 - show plan view 157
 - workload application 117
 - z/OS virtual workstation 77
 - z/OS workstation 76
- creating job definition
 - oslc automation, prerequisite steps 90
 - oslc provisioning, prerequisite steps 90
 - scp, prerequisite steps 89
- critical job
 - high risk status 71
 - monitoring 69
 - monitoring scenario
 - by using workload service assurance 173
 - planning 71
 - potential risk status 71
 - processing 69
 - tracking 71
- critical path
 - calculation of 69
- cross dependencies 49, 55
 - creating 99
- current plan 60
- customize
 - news notification beacon 25
- customizing
 - console 21
 - job stream
 - scenario 171
 - news notification beacon 185
 - portfolio 21
 - secondary queries 23
 - startup page 22
 - task 22
- cyclic periods
 - definition 39
 - examples 40

D

- daily run cycle 41
- database
 - definition 59
 - object properties 221
- database data extract 84, 91

- database data validation 84, 91
- database objects
 - creating 79
 - editing 79
 - from the Details view 82
 - from the Graphical view 82
 - from the Quick Open view 81
 - from the Working List view 80
- file 50
- job 38
- listing 114
- prompt 50
- resource 50
- variable table 58
- workstation class 57
- database operations
 - sample JSDL files 84, 91
- database repository
 - changing user 18
- database stored procedure
 - database jobs
 - sample JSDL files 84, 91
 - sample JSDL files 84, 91
- date syntax Cognos
 - prompt type 94
- DB2 repository
 - changing user 18
- defining
 - temporary plan 156
- defining workload application 117
- dependencies
 - adding 98
 - condition 49, 51
 - cross 49, 55
 - definition 48
 - external 48
 - file 50
 - internal 48
 - internetwork 49
 - prompt 50
 - removing 99
 - resource 50
 - using
 - to control job processing 149
 - to control job stream processing 149
- designing
 - scheduling environment 75
- Details view 216
 - editing
 - database objects 82
- disabling
 - news notification beacon 24, 185
- distributed workstation
 - creating 75
 - editing 77
- domain 34
 - creating 78
 - monitoring 129
- domain manager 31
 - attributes 75, 77
 - configuring 75, 77
- domains
 - monitoring 125
- dynamic agent 33
- dynamic agent workstation 73
- dynamic critical path 68

- dynamic domain manager
 - attributes 75, 77
 - configuring 75, 77
- dynamic pool 34, 73
 - creating 79
- dynamic scheduling 33, 34
- Dynamic Workload Broker
 - creating objects 107
- Dynamic Workload Console
 - accessibility ix
 - defining supported agents job 86, 88
 - getting started 1
 - troubleshooting 181
- dynamic workstations 33, 34

E

- editing
 - database objects 79
 - from the Details view 82
 - from the Graphical view 82
 - from the Quick Open view 81
 - from the Working List view 80
 - distributed workstation 77
 - event rules 113
 - objects in the database 75
- education x
- enabling
 - news notification beacon 24
- engine
 - associating a plan to 156
 - connection 63
- event
 - application monitoring 64
 - definition 64
 - generic 64
 - schemas for 233
 - management 64
 - task 131
 - objects related 64
 - properties 232
 - rules 66
- event rule
 - activation status 233
 - creating 109
 - definition
 - listing 116
 - monitoring 145
- event rules
 - editing 113
 - managing 113
- examples
 - SQL report 229
- exclusive run cycle 41
- executable jobs 84, 91
- exporting
 - settings 15
- expression
 - complex 225
 - regular 225, 226
- extended agent workstation 32, 73
- external dependencies 48

F

- fault-tolerant agent 32
 - attributes 75, 77
 - configuring 75, 77
- favorite
 - adding task to 23
- feed 26, 185
 - send notification 26
- file
 - definition 50
 - monitoring 141
 - monitoring event 64
- file transfer jobs
 - sample JSDL files 84, 91
- file transfer operations
 - sample JSDL files 84, 91
- files
 - monitoring 125
- FNCJSI
 - preproduction plan 62
- forecast plan 61
 - generating 157

G

- generating
 - forecast plan 157
 - report
 - Custom SQL 170
 - Job Run History 166
 - Job Run Statistics 165
 - Plan 168
 - Workstation Workload Runtimes 168
 - Workstation Workload Summary 167
 - trial plan 157
- generic
 - action 65
 - event 64
- generic Java job
 - template 84, 91
- generic Web service call
 - template 84, 91
- global prompt 50
- global settings
 - customizing 27
- glossary ix
- graphical 218
- Graphical view
 - editing
 - database objects 82
 - in the plan 210
- groups 196

H

- high risk status 71
- holidays calendar 41
- hot list 68
 - addition of jobs 70

I

- IBM i jobs 84, 91
 - AS400 jobs 84, 91
- impact view 212
- importing
 - settings 15
- inclusive run cycle 41
- installation
 - Dynamic Workload Console 7, 11
- installing
 - Dynamic Workload Console 7, 11
- internal dependencies 48
- internal status
 - distributed job stream 204
 - distributed jobs 200
 - z/OS job stream 205
 - z/OS jobs 202
- internetwork dependencies 49

J

- J2EE jobs 84, 91
- Java jobs
 - sample JSDL files 84, 91
- Java operations
 - sample JSDL files 84, 91
- JnextPlan 59
- job
 - ad hoc
 - setting properties for 162
 - submitting 161
 - adding to a job stream 94
 - addition to the hot list 70
 - controlling process
 - by using dependencies 149
 - by using job confirmation 152
 - by using job priority 151
 - by using job recovery actions 152
 - by using limits 151
 - by using time restrictions 150
 - by using workstation fence 151
 - critical
 - monitoring 69, 134
 - processing 69
 - risk level 136
 - definition 38
 - using recovery options in 222
 - internal status 200, 202
 - listing 115
 - monitoring 132, 137
 - monitoring scenario 175
 - predefined
 - setting properties for 162
 - submitting 161
 - return codes 229
 - statistics
 - return codes 229
 - status 123
 - description 199, 201
 - mapping 201, 202
 - time range 230
 - workload broker 58
- job definition
 - creating 84, 91
 - submit 97
 - submitting 97

- job definition (*continued*)
 - workload broker 58
- job stream
 - adding a job to 94
 - controlling process
 - by using dependencies 149
 - by using job priority 151
 - by using limits 151
 - by using time restrictions 150
 - by using workstation fence 151
 - customizing scenario 171
 - internal status 204, 205
 - listing 115
 - monitoring 138, 140
 - predefined
 - setting properties for 162
 - submitting 162
 - status
 - description 203, 205
 - mapping 204, 206
 - submit 97
 - submitting 97
- job stream definition
 - creating 83
- job stream view 212
- job types
 - template 84, 91
- job types with advanced options
 - sample JSDL files 84, 91
 - template 84, 91
- job-specific options 91

L

- limits
 - using
 - to control job processing 151
 - to control job stream processing 151
- listing
 - database objects 114
 - event rule definitions 116
 - job streams 115
 - jobs 115
 - workstations 117
- local prompt 50
- logical resource 50
- long term plan
 - preproduction plan 62

M

- managing
 - settings 15
 - settings repository 15, 19
- master domain manager 31
- message history 222
- monitor task 125
- monitoring
 - critical jobs 134
 - domain 129
 - domains 125
 - event rules 145
 - files 125, 141
 - job streams 125, 138
 - job streams on multiple engines 140

- monitoring (*continued*)
 - jobs 125, 132
 - jobs on multiple engines 137
 - scenario 175
 - objects in the plan 123
 - operator messages 148
 - plan 123
 - prompts 125, 143
 - resources 125, 144
 - scheduling environment 127
 - triggered actions 147
 - workload 130
 - workstation 125, 127
 - z/OS critical jobs
 - by using workload service assurance 173
- monitoring task
 - query line 125
- monthly run cycle 41
- MSSQL jobs 84, 91
- multiple engines
 - monitor job streams 140
 - monitor jobs 137
 - monitor jobs scenario 175
- multiple-domain network 35

N

- named prompt 50
- network
 - multiple-domain 35
 - single-domain 35
- new executors
 - template 84, 91
- new plug-ins 84, 91
 - template 84, 91
- news notification beacon 24
 - customize 185
 - disabling 24, 185
 - enabling 24
 - internal communications 26
 - recipients 25
- non-cyclic periods
 - examples 40
- noncyclic periods
 - definition 40
- notification
 - news
 - customizing 25
 - disabling 24, 185
 - enabling 24, 185
 - sending 26
- notification action 65

O

- object
 - creating 79
 - editing 79
 - from the Details view 82
 - from the Graphical view 82
 - from the Quick Open view 81
 - from the Working List view 80
 - listing 114
 - related event 64
- offset-based run cycle 41

- operational action 65
- operator instruction
 - definition 47
- operator message
 - monitoring 148
- option
 - conditional recovery job 54
 - recovered by condition 54
- oslc automation job definition
 - prerequisite steps 90
- oslc provisioning job definition
 - prerequisite steps 90

P

- parameter
 - definition 48
- path
 - critical
 - calculation of 69
- period
 - cyclic 39
 - definition 39
 - noncyclic 40
- physical resource 50
- plan
 - archived 61
 - associating
 - to engine 156
 - to task 155
 - current 60
 - forecast 61
 - generating 157
 - monitoring 123
 - monitoring objects 123
 - preproduction 60
 - production 60
 - progress 124
 - Symnew 60
 - temporary
 - defining 156
 - trial 61
 - generating 157
 - working
 - selecting 155
- plan extractor
 - regular expressions supported 226
- plan view 211
 - showing 157
 - task 132
- planning
 - critical job 71
- pool 34
 - creating 79
- portfolio
 - console 1
 - customizing 21
- potential risk status 71
- predecessor
 - successor 62
- preproduction plan 60
 - description 62
 - FNCJSI 62
 - long term plan 62
- production
 - plan 59, 60
 - process 59

- production (*continued*)
 - submitting workload in 161
- promotion 69
- prompt
 - abend 50
 - ad-hoc 50
 - global 50
 - local 50
 - monitoring 143
 - named 50
 - recovery 50
- prompt type Cognos
 - date syntax 94
 - time stamp syntax 94
 - time syntax 94
- prompts
 - monitoring 125
- properties
 - setting 83, 221
- Provisioning job definition
 - prerequisite steps 89
- publications ix
 - accessing 183

Q

- queries
 - secondary
 - customizing 23
- query line
 - monitoring task 125
- Quick Open view 215
 - editing
 - database objects 81

R

- recovered by condition option 54
- recovery options
 - using in job definitions 222
- recovery prompt 50
- regular expressions 225, 226
- remote engine workstation 33, 49, 55
- removing
 - dependency 99
- report
 - Custom SQL
 - generating 170
 - definition 66
 - format 67
 - header 67
 - Job Run History
 - generating 166
 - Job Run Statistics
 - generating 165
 - Plan
 - generating 168
 - Workstation Workload Runtimes
 - generating 168
 - Workstation Workload Summary
 - generating 167
- repository
 - changing user 18
 - settings 15, 16, 17, 19
- resource
 - logical 50

- resource (*continued*)
 - monitoring 144
 - physical 50
- resources
 - monitoring 125
- return codes
 - job statistics grouped by 229
 - jobs grouped by 229
- risk level 136
- rule-based run cycle 41
- run cycle
 - daily 41
 - exclusive 41
 - inclusive 41
 - monthly 41
 - offset-based 41
 - preview 221
 - rule-based 41
 - simple 41
 - weekly 41
 - yearly 41

S

- SAP
 - dynamic agent 73
 - dynamic pool 73
 - extended agent 73
 - Tivoli Workload Scheduler for 73
 - z-centric agent 73
- scalable vector graphics 221
- scenario 171
- scheduling environment
 - designing 75
 - monitoring 127
- schemas for generic events 233
- secondary queries
 - customizing 23
- selecting
 - working plan 155
- Self-Service Catalog
 - auditing 193
- Self-Service Dashboards
 - auditing 193
- setting
 - properties 83, 221
 - for ad hoc jobs 162
 - for predefined job streams 162
 - for predefined jobs 162
- settings
 - changing repository 16, 19
 - exporting 15
 - importing 15
 - managing 15
 - repository 15, 16, 19
 - saving in database 16, 19
 - sharing repository 17
- settings repository
 - managing 15, 19
 - sharing 17
- shadow job 38, 49
- sharing
 - settings repository 17
- showing
 - plan view 157
- simple run cycle 41
- single-domain network 35

- specific job types
 - sample JSDL files 84, 91
- SQL report
 - examples 229
- SSL communication options 198
- standard agent 32
- starting
 - console 1
- startup page
 - customizing 22
- status description
 - distributed job stream 203
 - distributed jobs 199
 - z/OS job stream 205
 - z/OS jobs 201
- status mapping
 - distributed job stream 204
 - distributed jobs 201
 - z/OS job stream 206
 - z/OS jobs 202
- submit
 - job 97
 - job stream 97
- submitting
 - ad hoc jobs 161
 - job 97
 - job stream 97
 - predefined job streams 162
 - predefined jobs 161
 - workload in production 161
- successor
 - predecessor 62
- supported agents job defining with
 - Dynamic Workload Console 86, 88
- Symnew plan 60
- Symphony file 59, 72
- syntax for date Cognos
 - prompt type 94
- syntax for parameterized filter Cognos
 - parameterized filter 94
 - prompt type 94
- syntax for time Cognos
 - parameterized filter 94
 - prompt type 94
- syntax for time stamp Cognos
 - prompt type 94

T

- task
 - adding
 - to favorite bookmarks 23
 - associating a plan to 155
 - customizing 22
 - event management 131
 - generate Custom SQL report 170
 - generate Job Run History report 166
 - generate Job Run Statistics report 165
 - generate Plan report 168
 - generate Workstation Workload Runtimes report 168
 - generate Workstation Workload Summary report 167
 - list event rule definitions 116
 - monitor 125
 - monitor critical jobs 134

- task (*continued*)
 - monitor domains 129
 - monitor event rules 145
 - monitor files 141
 - monitor job streams 125, 138
 - monitor job streams on multiple engines 140
 - monitor jobs 125, 132
 - monitor jobs on multiple engines 137
 - monitor operator messages 148
 - monitor prompts 143
 - monitor resources 144
 - monitor triggered actions 147
 - monitor workstations 125, 127
 - plan view 132
 - show plan view 157
- technical training x
- time range 230
- time restrictions
 - using
 - to control job processing 150
 - to control job stream processing 150
- time stamp syntax Cognos
 - prompt type 94
- time syntax Cognos
 - prompt type 94
- Tivoli Dynamic Workload Console
 - getting started 1
- Tivoli technical training x
- Tivoli Workload Scheduler for z/OS
 - agent 33
- tracking
 - critical job 71
- training
 - technical x
- trial plan 61
 - generating 157
- triggered actions
 - monitoring 147
- troubleshooting
 - Dynamic Workload Console 181

U

- usage scenario 171
- user
 - change password 121
- users 196
- using
 - news notification beacon 24

V

- variable information
 - using into action properties 231
- variable table 58
- view
 - details 216
 - graphical 218
 - impact 212
 - job stream 212
 - plan 211
- virtual workstation 33

W

- Web service jobs
 - sample JSDL files 84, 91
- weekly run cycle 41
- working list 214
- Working List view
 - editing
 - database objects 80
- working plan
 - monitoring 155
- workload
 - monitoring 130
 - submitting in production 161
- workload application 117
- workload broker
 - agent workstation 33
 - job definition 58
- workload designer 214
- workload service assurance 68
 - scenario 173
- workstation
 - backup master domain manager 31
 - class 57
 - creating 75, 76, 77
 - definition 31
 - domain manager 31
 - dynamic agent 33, 73
 - dynamic pool 34
 - editing 77
 - extended agent 32, 73
 - fault-tolerant agent 32
 - master domain manager 31
 - monitoring 125, 127
 - pool 34
 - remote engine 33, 55
 - standard agent 32
 - Tivoli Workload Scheduler for z/OS
 - agent 33
 - virtual 33
 - workload broker agent 33
 - z-centric agent 73
- workstation fence
 - using
 - to control job processing 151
 - to control job stream processing 151
- workstations
 - listing 117

Y

- yearly run cycle 41

Z

- z-centric agent workstation 73
- z/OS virtual workstation
 - creating 77
- z/OS workstation
 - creating 76



Product Number: 5698-WSH, 5698-T08

Printed in USA