

IBM Tivoli Workload Scheduler



Troubleshooting Guide

Version 9 Release 1

IBM Tivoli Workload Scheduler



Troubleshooting Guide

Version 9 Release 1

Note

Before using this information and the product it supports, read the information in “Notices” on page 221.

This edition applies to version 9, release 1, 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 2001, 2013.**

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

Contents

Figures	ix
--------------------------	-----------

Tables	xi
-------------------------	-----------

About this guide	xiii
-----------------------------------	-------------

What is new in this release	xiii
What is new in this release for troubleshooting	xiii
Who should read this publication.	xiii
Publications	xiii
Accessibility	xiv
Tivoli technical training	xiv
Support information	xiv

Chapter 1. Getting started with troubleshooting	1
--	----------

Where products and components are installed	1
Finding out what has been installed in which Tivoli Workload Automation instances.	4
Built-in troubleshooting features.	6
Keeping up-to-date with the latest fix packs	7
Upgrading your whole environment	7

Chapter 2. Logging and tracing.	9
--	----------

Quick reference: how to modify log and trace levels	9
Difference between logs and traces	13
Tivoli Workload Scheduler logging and tracing using CCLog	14
Engine log and trace file locations	14
Engine log and trace file switching	15
Engine log and trace customization	15
Engine log and trace performance	18
Engine Log Analyzer	18
Dynamic Workload Console log and trace files	30
Activating and deactivating traces in Dynamic Workload Console	31
Dynamic workload scheduling log and trace files.	32
Activating logs for Job Brokering Definition Console.	32
Dynamic agent log and trace files	33
Configuring trace properties when the agent is running.	34
Log and trace files for the application server	37
Setting the traces on the application server for the major Tivoli Workload Scheduler processes	37
Log files for the command line client.	39

Chapter 3. Capturing data in the event of problems	41
---	-----------

Data capture utility.	41
When to run the utility	41
Prerequisites	42
Command and parameters	42
Tasks	43
Data collection	44

Data structure	47
First failure data capture (ffdc)	50
Creating a core dump of the application server	51

Chapter 4. In-Flight Trace facility for engine	53
---	-----------

In-Flight Trace configuration file	55
Changing the configuration	56
Configuration file syntax	56
In-Flight Trace command: xcli	58
Selecting programs, segments, and products	59
xcli command syntax	59
xcli messages.	65

Chapter 5. Auditing facilities	69
---	-----------

Database and plan audit	69
How audit works	69
Enabling the audit feature	70
Audit log header format	70
Audit log body format.	71
Dynamic workload scheduling audit	75
Log file specifications	78
How to perform queries on log files	80
Examples	81

Chapter 6. Troubleshooting performance issues	85
--	-----------

Chapter 7. Troubleshooting networks	87
--	-----------

Network recovery	87
Initialization problems.	87
Network link problems	88
Replacement of a domain manager	89
Replacement of a master domain manager	89
Other common network problems	89
Using SSL, no connection between a fault-tolerant agent and its domain manager	90
After changing SSL mode, a workstation cannot link	90
In a configuration with a firewall, the start and stop remote commands do not work	91
Remote command job fails to connect to remote computer	91
The domain manager cannot link to a fault-tolerant agent	91
Changes to the SSL keystore password prevent the application server from starting	92
Agents not linking to master domain manager after first JnextPlan on HP-UX	92
Fault-tolerant agents not linking to master domain manager	93
The dynamic agent cannot be found from Dynamic Workload Console	94
Submitted job is not running on a dynamic agent	94

Job status of a submitted job is continually shown as running on dynamic agent	94
Network performance	95
AWSITA245E or AWSITA104E - Agent is down but jobmanager is running.	95

Chapter 8. Troubleshooting common engine problems 97

Composer problems	97
Composer gives a dependency error with interdependent object definitions	97
The display cpu=@ command does not work on UNIX	98
Composer gives the error "user is not authorized to access server".	98
The deletion of a workstation fails with the "AWSJOM179E error	99
When using the composer add and replace commands, a Job Scheduler has synchronicity problems	99
JnextPlan problems	99
JnextPlan fails to start	100
JnextPlan fails with the database message "The transaction log for the database is full.".	100
JnextPlan fails with a Java out-of-memory error	101
JnextPlan fails with the DB2 error like: nullDSRA0010E	101
JnextPlan fails with message AWSJPL017E.	102
JnextPlan is slow	102
A remote workstation does not initialize after JnextPlan	102
A workstation does not link after JnextPlan	103
A job remains in "exec" status after JnextPlan but is not running.	103
A change in a resource quantity in the database is not also implemented in the plan after JnextPlan	104
On SLES8, after the second JnextPlan, an agent does not link	105
CreatePostReports.cmd, or Makeplan.cmd, or Updatestats.cmd hang on Windows operating systems	105
Conman problems	106
On Windows, the message AWSDEQ024E is received	106
Conman on a SLES8 agent fails because a library is missing	107
Duplicate ad-hoc prompt number	108
Submit job streams with a wildcard loses dependencies	108
Fault-tolerant agent problems	109
A job fails in heavy workload conditions	109
Batchman, and other processes fail on a fault-tolerant agent with the message AWSDEC002E	110
Fault-tolerant agents unlink from mailman on a domain manager	110
Dynamic agent problems	111
The dynamic agent cannot contact the server	111
V8.5.1 fault-tolerant agent with dynamic capabilities cannot be registered	111

Error message AWKDBE009E is received	112
Error message AWSITA104E is received.	112
Problems on Windows	113
Interactive jobs are not interactive using Terminal Services	113
The Tivoli Workload Scheduler services fail to start after a restart of the workstation	113
The Tivoli Workload Scheduler for user service (batchup) fails to start	114
An error relating to impersonation level is received	115
Extended agent problems	116
The return code from an extended agent job is not recognized	116
Planner problems	116
There is a mismatch between Job Scheduler instances in the Symphony file and the preproduction plan	116
Planman deploy error when deploying a plug-in	117
An insufficient space error occurs while deploying rules.	117
UpdateStats fails if it runs more than two hours (message AWSJCO084E given).	117
The planman showinfo command displays inconsistent times	118
A bound z/OS shadow job is carried forward indefinitely	118
Problems with DB2	119
Timeout occurs with DB2	119
JnextPlan fails with the DB2 message "The transaction log for the database is full.".	119
The DB2 UpdateStats job fails after 2 hours	119
DB2 might lock while making schedule changes	120
Problems with Oracle.	121
JnextPlan fails with the database message "The transaction log for the database is full.".	121
You cannot do Oracle maintenance on UNIX after installation	122
Application server problems	122
The application server does not start after changes to the SSL keystore password	122
Timeout occurs with the application server	122
On windows, "java process already running" error, restarting the WebSphere Application Server	123
Event management problems	124
Troubleshooting an event rule that does not trigger the required action	124
Actions involving the automatic sending of an email fail	131
An event is lost	131
Expected actions not triggered following an event	131
Event rules not deployed after switching event processor	132
Event <i>LogMessageWritten</i> is not triggered	132
Deploy (D) flag not set after ResetPlan command used.	133
Missing or empty event monitoring configuration file	133
Events not processed in correct order	134

The stopeventprocessor or switcheventprocessor commands do not work	134
Event rules not deployed with large numbers of rules	134
Problem prevention with disk usage, process status, and mailbox usage	135
Problems using the "legacy" global options	135
Time zones do not resolve correctly with enLegacyStartOfDayEvaluation set	135
Dependencies not processed correctly when enLegacyId set	135
Managing concurrent accesses to the Symphony file	136
Scenario 1: Access to Symphony file locked by other Tivoli Workload Scheduler processes	136
Scenario 2: Access to Symphony file locked by stageman	136
StartApp Server problems	136
MakePlan problems	136
MakePlan fails to start	137
Unable to establish communication with the server on host - AWSBEH023E	137
The user "twuser" is not authorized to access the server on host - AWSBEH021E	137
The database is already locked - AWSJPL018E	137
An internal error has occurred - AWSJPL006E	137
The production plan cannot be created - AWSJPL017E	138
An internal error has occurred - AWSJPL074E	138
SwitchPlan problems	138
When SwitchPlan fails to start	138
The previous Symphony file and Symnew file have the same run number - AWSBHV082E	139
Create Post Reports	139
Update Stats problems	139
Miscellaneous problems	139
An error message indicates that a database table, or an object in a table, is locked	140
Command line programs (like composer) give the error "user is not authorized to access server"	141
The rmstdlist command gives different results on different platforms	141
The rmstdlist command fails on AIX with an exit code of 126.	141
Question marks are found in the stdlist	141
Deleting stdlist or one of its files when processes are still running	142
A job with a "rerun" recovery job remains in the "running" state	142
Job statistics are not updated daily	143
A Job Scheduler dependency is not added.	143
Incorrect time-related status displayed when time zone not enabled	143
Completed job or job stream not found.	143
Variables not resolved after upgrade.	144
Default variable table not accessible after upgrade	144
Local parameters not being resolved correctly	144

Log files grow abnormally large in mixed environment with version 8.4 or higher master domain manager and 8.3 or lower agents	145
Inconsistent time and date in conman and planman output	146
Deleting leftover files after uninstallation is too slow	146
Corrupted special characters in the job log from scripts running on Windows	147
Failover Cluster Command Interface deprecated	147
StartUp shows an error after upgrade	147

Chapter 9. Troubleshooting dynamic workload scheduling 149

How to tune the rate of job processing	149
Remote command job fails	151
On Windows 2012 interactive jobs are not visible on dynamic agents	152
Troubleshooting common problems	152
On AIX operating systems the concurrent submission of one hundred or more jobs on the same agent can result in a core dump or in a resource temporarily unavailable message.	152
Dynamic workload broker cannot run after the Tivoli Workload Scheduler database is stopped	153
Getting an OutofMemory exception when submitting a job	153
Getting an error exception when submitting a job on a fresh agent	153

Chapter 10. Troubleshooting Dynamic Workload Console problems 155

Troubleshooting connection problems	155
The engine connection does not work	156
Test connection takes several minutes before returning failure	157
Engine version and connection status not displayed.	158
Failure in testing a connection or running reports on an engine using an Oracle database	158
Connection error when running historical reports or testing connection from an external instance of WebSphere Application Server	159
Connection problem with the engine when performing any operation	159
Engine connection does not work when connecting to the z/OS connector (versions 8.3.x and 8.5.x).	160
Engine connection does not work when connecting to the z/OS connector V8.3.x or a distributed Tivoli Workload Scheduler engine V8.3.x	161
Engine connection does not work when connecting to distributed Tivoli Workload Scheduler engine V8.4 FP2 on UNIX.	162
WebSphere does not start when using an LDAP configuration	163
WebSphere Application Server problem when using an LDAP configuration with SSL authentication mechanism	164

Engine connection settings are not checked for validity when establishing the connection . . .	165	Some panels in Dynamic Workload Console might not be displayed correctly in Internet Explorer, version 8 and 9	176
LDAP account locked after one wrong authentication attempt	165	Web page error with Internet Explorer, version 9	177
Troubleshooting performance problems.	166	Dynamic Workload Console problems with Internet Explorer developer tools	177
With a distributed engine the responsiveness decreases overtime	166	Troubleshooting problems with graphical views	177
Running production details reports might overload the distributed engine	166	Language-specific characters are not correctly displayed in graphical views	177
A "java.net.SocketTimeoutException" received	166	Plan View limit: maximum five users using the same engine.	178
Troubleshooting user access problems	168	Job stream no longer displayed in job stream view	178
Wrong user logged in when using multiple accesses from the same system	168	Graphical view and dashboard stop responding on shared WebSphere Application Server on Solaris.	178
Unexpected user login request after having configured to use Single Sign-On.	168	AWSITA122E or AWKRAA209E error while working with jobs in the Workload Designer . .	178
Single Sign-On authentication doesn't work anymore from Dynamic Workload Console with master version 8.5.1 and master upgraded to version 8.6.0.	169	Troubleshooting problems with database	179
Authentication problem: the administrative user cannot see the portfolio	169	Communication failure with DB2 when working with the Dynamic Workload Console on a Red Hat Enterprise Linux (RHEL) Version 5.6 system	179
Authentication problem when opening the Workload Designer	170	Import preferences fails	179
Troubleshooting problems with reports	170	Troubleshooting configuration problems	180
The output of a report run on Job Statistics View shows -1 in the Average CPU Time and Average Duration fields	171	IHS load balancing configuration for Dynamic Workload Console cluster fails.	180
The validate command running on a custom SQL query returns the error message AWSWUI0331E.	171	Troubleshooting other problems	182
The output of report tasks is not displayed in a browser with a toolbar installed	171	The deletion of a workstation fails with the "AWSJOM179E error	183
WSWUI0331E error when running reports on an Oracle database	172	Data not updated after running actions against monitor tasks results	183
CSV report looks corrupted on Microsoft Excel not supporting UTF8.	172	"Session has become invalid" message received	183
Insufficient space when running production details reports	172	Actions running against scheduling objects return empty tables	184
After Tivoli Workload Scheduler upgrades from version 8.3 to version 8.5 some fields in the output of reports show default values (-1, 0, unknown, regular).	173	Default tasks are not converted into the language set in the browser	184
Report error: the specified run period exceeds the historical data time frame	173	"Access Error" received when launching a task from the browser bookmark	185
Troubleshooting problems with browsers	174	If you close the browser window, processing threads continue in the background	185
Default tasks are not converted into the language set in the browser	174	The list of Available Groups is empty in the Enter Task Information window	185
"Access Error" received when launching a task from the browser bookmark	174	JVM failure when working with the Dynamic Workload Console on a Red Hat Enterprise Linux (RHEL) Version 5 system	186
If you close the browser window, processing threads continue in the background	175	Missing daylight saving notation in the time zone specification on Dynamic Workload Console 8.4 Fix Pack 1 and later	186
Unresponsive script warning with Firefox browser	175	A "java.net.SocketTimeoutException" received	186
Plan View panel seems to freeze with Internet Explorer version 7.	175	Blank page displayed (in High availability disaster recovery configuration)	186
Blank page displayed (in High availability disaster recovery configuration)	176	Some panels in Dynamic Workload Console might not be displayed correctly	187
Workload Designer does not show on foreground with Firefox browser	176	Exceptions might not be displayed in Language-specific in the Dynamic Workload Console	187
		Extraneous exception logged in SystemOut . .	188
		Filtering task results might not work as expected	188
		Sorting task results might not work as expected	189

Chapter 11. Troubleshooting workload service assurance	191		
Components involved in workload service assurance.	191	Recovering using the backup master domain manager	205
Exchange of information	192	Recover using the logman and ResetPlan commands	207
Common problems with workload service assurance.	192	Recovering the plan from the latest archived plan	208
Critical start times not aligned.	192	Recovery procedure on a fault-tolerant agent or lower domain manager	210
Critical start times inconsistent	193	Recovery procedure on a fault-tolerant agent with the use of the resetFTA command	211
Critical network timings change unexpectedly	193		
A critical job is consistently late	193	Appendix A. Support information	213
A high risk critical job has an empty hot list	194	IBM Support Assistant	213
		Searching knowledge bases.	214
Chapter 12. Troubleshooting the fault-tolerant switch manager	195	Search the local information center	214
Event counter	195	Search the Internet	214
Ftbox	196	Obtaining fixes	215
Troubleshooting link problems	196	Receiving support updates	215
Common problems with the backup domain manager	200	Contacting IBM Software Support	216
The Symphony file on the backup domain manager is corrupted.	201	Determine the business impact	217
Processes seem not to have been killed on previous UNIX domain manager after running switchmgr	201	Describe problems and gather information	217
In a scenario involving more than one switchmgr command, agent cannot relink	201	Submit problems	217
Chapter 13. Synchronizing the database with the Symphony file	203	Appendix B. Date and time format reference - strftime	219
Chapter 14. Corrupt Symphony file recovery	205	Notices	221
Recovery procedure on a master domain manager	205	Trademarks	222
		Index	225

Figures

1. ACCT_FS has not linked 197
2. Example output for **conman sc @!@** run on the master domain manager 198
3. Example output for **conman sc** run on the domain manager 198
4. Example output for **conman sc** run on the unlinked workstation 199
5. Example output for **conman sc @!@** run on the unlinked workstation 200
6. Example output for **ps -ef | grep writer** run on the unlinked workstation 200

Tables

1. Where to find other troubleshooting material	1	11. Elements in Outcome type	79
2. Difference between logs and traces.	14	12. Elements in UserInfoList type	80
3. Locations of log files and trace files	30	13. Elements in UserInfo type.	80
4. Locations of log and trace files	32	14. Job processing status to queue jobs for dispatching	150
5. Collected data structure on UNIX	47	15. Default settings for new job run statistic reports.	173
6. Collected data structure on Windows	48	16. Default settings for new job run history reports.	173
7. Auditable event properties	77	17. strftime date and time format parameters	219
8. Elements in Action type	78		
9. Elements in ObjectInfoList type	78		
10. Elements in ObjectInfo type	78		

About this guide

Gives useful information about the guide, such as what it contains, who should read it, what has changed since the last release, and how to obtain training and support.

IBM® Tivoli® Workload Scheduler: Troubleshooting provides information about troubleshooting IBM Tivoli Workload Scheduler and its components.

What is new in this release

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

For information about the APARs that this release addresses, see the Tivoli Workload Scheduler Release Notes at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27038323> and the Dynamic Workload Console Release Notes at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27038328>.

What is new in this release for troubleshooting

This section describes what has changed in this release with regard to troubleshooting since version 8.5.1.

Who should read this publication

This publication is designed to help users deal with any error situations they encounter while working with Tivoli Workload Scheduler. The publication includes targeted troubleshooting information about some specific activities and solutions to problems that you might encounter while running the product.

Some of these solutions need an expert user of Tivoli Workload Scheduler to resolve them, while others require the expertise of an expert systems programmer, who has a reasonable understanding of the Tivoli Workload Scheduler infrastructure and its inter-component interactions.

Publications

Full details of Tivoli Workload Scheduler publications can be found in *Tivoli Workload Automation: Publications*. This document also contains information about the conventions used in the publications.

A glossary of terms used in the product can be found in *Tivoli Workload Automation: Glossary*.

Both of these are in the Information Center as separate publications.

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

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

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

Support information

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. Getting started with troubleshooting

Gives an overview of what troubleshooting information is contained in this publication, and where to find troubleshooting information which is *not* included.

This publication gives troubleshooting information about the Tivoli Workload Scheduler *engine*. The engine comprises the components of Tivoli Workload Scheduler that perform the workload scheduling activities, together with the command line by which they can be controlled.

Troubleshooting for other Tivoli Workload Scheduler activities, products, and components can be found in their relevant publications, as follows:

Table 1. Where to find other troubleshooting material

Activity, Product, or Component	Publication
Installation, upgrade, and uninstallation of Tivoli Workload Scheduler components and the Dynamic Workload Console	<i>Tivoli Workload Scheduler: Planning and Installation Guide, SC32-1273</i>
Limited fault-tolerant agents for IBM i	<i>Tivoli Workload Scheduler: Limited Fault-tolerant Agent for IBM i, SC32-1280</i>
Tivoli Workload Scheduler for z/OS®	<i>Tivoli Workload Scheduler for z/OS: Diagnosis Guide and Reference, SC32-1261</i> <i>Tivoli Workload Scheduler for z/OS: Messages and Codes, SC32-1267</i>
Tivoli Workload Scheduler for Applications	<i>Tivoli Workload Scheduler for Applications: User's Guide, SC32-1278</i>
Tivoli Workload Scheduler for Virtualized Data Centers	<i>Tivoli Workload Scheduler for Virtualized Data Centers: User's Guide, SC32-1454</i>

Many of the procedures described in this publication require you to identify a file in the installation path of the product and its components. However, they can have more than one installation path, as described in "Where products and components are installed."

Where products and components are installed

Describes where the Tivoli Workload Scheduler products and components are installed.

This section commences by briefly introducing Tivoli Workload Automation and explaining how this concept impacts the installed structure of Tivoli Workload Scheduler.

Tivoli Workload Automation

Tivoli Workload Automation is the name of a family of products and components, which includes the following:

- Tivoli Workload Scheduler
- Tivoli Workload Scheduler for z/OS
- Tivoli Workload Scheduler for Applications
- Dynamic Workload Console
- Tivoli Workload Scheduler for Virtualized Data Centres
- Tivoli Workload Scheduler Plug-in for Informatica PowerCenter

Many Tivoli Workload Scheduler components are installed in what is called a *Tivoli Workload Automation instance*.

Installation paths

TWA_home installation path

As described above, many of the components are installed in a Tivoli Workload Automation instance. Although this is a notional structure it is represented on the computer where you install Tivoli Workload Automation components by a common directory referred to in the documentation as *TWA_home*. The path of this directory is determined when you install a Tivoli Workload Scheduler component for the first time on a computer. You have the opportunity to choose the path when you make that first-time installation, but if you accept the default path, it is as follows:

Linux /opt/IBM/TWA<n>

UNIX /opt/ibm/TWA<n>

Windows

C:\Program Files\IBM\TWA<n>

where <n> is an integer value ranging from <null> for the first instance installed, 1 for the second, and so on.

This path is called, in the publications, *TWA_home*. For details about the directories created outside of *TWA_home*, see *Tivoli Workload Scheduler: Planning and Installation*.

Tivoli Workload Scheduler installation path

You can install more than one Tivoli Workload Scheduler component (master domain manager, backup master domain manager, domain manager, or backup domain manager) on a system, but each is installed in a separate instance of .Tivoli Workload Automation, as described above.

The installation path of Tivoli Workload Scheduler is:

TWA_home/TWS

Tivoli Workload Scheduler agent installation path

The agent also uses the same default path structure, but has its own separate installation directory:

TWA_home/TWS/ITA/cpa

Note: The agent also installs some files outside this path. If you have to share, map, or copy the agent files (for example, when configuring support for clustering) share, map, or copy these files as well:

UNIX and Linux operating systems

```
/etc/teb/teb_tws_cpa_agent_<TWS_user>.ini
/opt/IBM/CAP/EMICPA_default.xml
/etc/init.d/tebctl-tws_cpa_agent_<TWS_user>
  (on Linux and Solaris)
/etc/rc.d/init.d/tebctl-tws_cpa_agent_<TWS_user>
  (on AIX)
/sbin/init.d/tebctl-tws_cpa_agent_<TWS_user>
  (on HP-UX)
```

Windows operating systems

```
%windir%\teb\teb_tws_cpa_agent_&lt;tws_user>.ini
%ALLUSERSPROFILE%\Application Data\ibm\CAP\EMICPA_default.xml
```

The agent uses the following configuration files which you might need to modify:

JobManager.ini

This file contains the parameters that tell the agent how to run jobs. You should only change the parameters if advised to do so in the Tivoli Workload Scheduler documentation or requested to do so by IBM Software Support. Its path is:

```
TWA_home/TWS/ITA/cpa/config/JobManager.ini
```

ita.ini This file contains parameters which determine how the agent behaves. Changing these parameters may compromise the agent functionality and require it to be reinstalled. You should only change the parameters if advised to do so in the Tivoli Workload Scheduler documentation or requested to do so by IBM Software Support. Its path is:

```
TWA_home/TWS/ITA/cpa/ita/ita.ini
```

Installation path for files giving the dynamic scheduling capability

The files that give the dynamic scheduling capability are installed in the following path:

```
TWA_home/TDWB
```

Dynamic Workload Console installation path

The Dynamic Workload Console can be installed in the path of your choice, but the default installation path is as follows:

On Windows

```
C:\Program Files\IBM\TWAUI
```

On UNIX

```
/opt/IBM/TWAUI
```

The WebSphere Application Server installation path

The WebSphere Application Server is automatically installed when you create a new *Tivoli Workload Automation instance*. You can specify any path for the installation. The default installation path is:

```
TWA_home/WAS
```

For the Dynamic Workload Console: C:\Program Files\IBM\JazzSM

The command line client installation path

The command line client is installed outside all *Tivoli Workload Automation instances*. Its default path is:

UNIX /opt/ibm/TWS/CLI

Windows

```
C:\Program Files\IBM\TWS\CLI
```

The application server tools installation path

Because the WebSphere Application Server is not supplied with an administration GUI, many of its administration tasks are performed by running tools supplied with Tivoli Workload Scheduler, that perform the required configuration changes. These tools are known as the *wastools*, and are installed in:

`TWA_home/wastools`

However, the information above supplies only the *default* paths. To determine the actual paths of products and components installed in your Tivoli Workload Automation instances, see “Finding out what has been installed in which Tivoli Workload Automation instances”

Finding out what has been installed in which Tivoli Workload Automation instances

How to identify which Tivoli Workload Scheduler components are installed on a computer.

If you are not the installer of Tivoli Workload Scheduler and its components, you might not know what components have been installed, and in which instances of Tivoli Workload Automation. Follow this procedure to find out:

1. Access the following directory:

`/etc/TWA`

`/etc/TWA`

`%windir%\TWA`

2. List the contents of the directory. Each Tivoli Workload Automation instance is represented by a file called: `twainstance<instance_number>.TWA.properties`. These files are deleted when all the products or components in an instance are uninstalled, so the number of files present indicates the number of valid instances currently in use.
3. Open a file in a text viewer.

Attention: Do not edit the contents of this file, unless directed to do so by IBM Software Support. Doing so might invalidate your Tivoli Workload Scheduler environment.

The contents are similar to this:

```
TWS_version=9.1.0.0
DB2_basePath=/home/db2inst1/sqllib
DB2_IS_SERVER=TRUE
EWas_basePath=/opt/IBM/WebSphere/AppServer
DB2_INSTANCE_PORT=50000
TWS_counter=1
EWas_counter=1
TWA_path=/opt/tws/tws
TWS_server_name=bvtserver
DB2_ADMINISTRATOR_NAME=db2inst1
TWS_instance_type=MDM
EWas_profile_path=/opt/tws/tws/Appserver/profiles/TWSProfile
EWas_node_name=TWSNode
TWS_basePath=/opt/tws/tws/TWS
EWas_user=tws
EWas_cell_name=TWSCell
EWas_version=8.5.0.1
DB2_version=10.1.0.0
EWas_server_name=server1
EWas_update_installer_dir=
```

```
TWS_LAST_COMMITTED_LEVEL_KEY=9.1.0.00
TWS_user_name=twS
TWS_FIX_LIST_KEY=
DB2_INSTANCE_NAME=db2inst1
DB2_counter=1
TWA_componentList=TWS,EWas,DB2
EWas_isc_version_key=8.5.0.1
EWas_profile_name=BVTProfile
EWas_service_name=IBMWAS85Service - twS
```

The important keys to interpret in this file are:

TWA_path

This is the base path, to which the installation added one or more of the following directories, depending on what was installed:

TWS Where the Tivoli Workload Scheduler component is installed

TDWC Where the Dynamic Workload Console is installed

eWAS Where the WebSphere Application Server is installed

wastools

Where the tools that you use to configure WebSphere Application Server are installed

TWA_componentList

Lists the components installed in the instance of Tivoli Workload Automation

TWS_counter

Indicates if a Tivoli Workload Scheduler component is installed in this instance of Tivoli Workload Automation (when the value=1)

TWS_instance_type

Indicates which component of Tivoli Workload Scheduler is installed in this instance:

MDM Master domain manager

BKM Backup master domain manager

FTA Agent or domain manager

TDWC_counter

Indicates if an instance of Dynamic Workload Console is installed in this instance of Tivoli Workload Automation (when the value=1)

EWas_counter

Indicates how many applications are installed in this instance of Tivoli Workload Automation that access the WebSphere Application Server

TWS_user_name

The ID of the <TWS_user> of the Tivoli Workload Scheduler component.

The only component of Tivoli Workload Scheduler which is installed in a Tivoli Workload Automation instance, but which is not explicitly indicated here, is the Connector. To determine if it has been installed, look at the following combinations of keys:

Agent installed with no Connector

```
TWS_counter=1
TWS_instance_type=FTA
TWA_componentList=TWS
```

Agent installed with Connector

```
TWS_counter=1
EWas_counter=1
TWS_instance_type=FTA
TWA_componentList=TWS,EWas
```

Agent installed with no Connector and Dynamic Workload Console

```
TWS_counter=1
EWas_counter=1
TWS_instance_type=FTA
TDWC_counter=1
TWA_componentList=TWS,EWas,TDWC
```

Agent installed with Connector and Dynamic Workload Console

```
TWS_counter=1
EWas_counter=2
TWS_instance_type=FTA
TDWC_counter=1
TWA_componentList=TWS,EWas,TDWC
```

Note: The only difference between these last two is that the EWas_counter is 2 instead of 1.

Built-in troubleshooting features

A list, brief description and links to more information on the tools and facilities which are built in to the product to facilitate troubleshooting.

Tivoli Workload Scheduler is supplied with the following features that assist you with troubleshooting:

- Informational messages that inform you of expected events.
- Error and warning messages that inform you of unexpected events.
- Message helps for the most commonly-occurring messages. See *Tivoli Workload Scheduler: Messages*.
- A logging facility that writes all types of messages to log files, which you use to monitor the progress of Tivoli Workload Scheduler activities. See “Tivoli Workload Scheduler logging and tracing using CCLog” on page 14.
- Various tracing facilities which record at varying levels of details the Tivoli Workload Scheduler processes for troubleshooting by IBM Software Support. See “Difference between logs and traces” on page 13 for more details.
- A facility to save a configurable level of log and tracing information in memory and then save all or part of this information to a single fully integrated file for troubleshooting by IBM Software Support. See Chapter 4, “In-Flight Trace facility for engine,” on page 53 for more details.
- A Log Analyzer that you use to read, analyze and compare log and some trace files. See “Engine Log Analyzer” on page 18.
- An auditing facility that provides an audit trail of changes to the Tivoli Workload Scheduler database and plan for use in both monitoring and troubleshooting. See the section about Auditing in the *Tivoli Workload Scheduler Administration*, for more details.
- A configuration snapshot facility that you can use for backup, and also which provides IBM Software Support with configuration information when unexpected events occur. See “Data capture utility” on page 41.

- A facility that automatically creates a First Failure Data Capture (ffdc) configuration snapshot if the failure of any of the key components can be detected by its parent component. See “First failure data capture (ffdc)” on page 50.
- An automatic backup mechanism of the Symphony file whereby each fault-tolerant agent and domain manager that receives a new Symphony file, automatically archives the previous Symphony to Symphony.last in the path <TWA_home/TWS/, so that a backup copy is always maintained. This permits viewing of the previous Symphony data in case there were any message updates on the job and job stream states that were lost between the agent and its master domain manager.

Keeping up-to-date with the latest fix packs

Reminds you that the best way to avoid problems is to apply fix packs

Tivoli Workload Scheduler fix packs contain fixes to problems that IBM, you, or other customers have identified. Install the latest fix pack when it becomes available, to keep the product up to date.

Upgrading your whole environment

When upgrading, although compatibility with previous version components is a feature of Tivoli Workload Scheduler, potential problems can be avoided by upgrading all components to the new level as quickly as possible.

To avoid problems, ensure that when you upgrade to a new version of Tivoli Workload Scheduler you do so across your whole environment.

The components of this version of Tivoli Workload Scheduler are compatible with components of many previous versions (see *Tivoli Workload Automation: Overview* for full details). However, running Tivoli Workload Scheduler in a mixed network increases the possibility of problems arising, because each new release of Tivoli Workload Scheduler not only adds functions, but also improves the stability and reliability of the various components. Try not to run in a mixed network for extended periods.

Chapter 2. Logging and tracing

Provides detailed information about logs and traces, and how to customize them and set the logging and tracing levels.

Information on the logging and tracing facilities of Tivoli Workload Scheduler, Dynamic Workload Console, and the WebSphere Application Server is described in these topics:

- “Quick reference: how to modify log and trace levels”
- “Difference between logs and traces” on page 13
- “Tivoli Workload Scheduler logging and tracing using CCLog” on page 14
- “Dynamic Workload Console log and trace files” on page 30
- “Dynamic workload scheduling log and trace files” on page 32
- “Dynamic agent log and trace files” on page 33
- “Log and trace files for the application server” on page 37
- “Log files for the command line client” on page 39

For details of the installation log files, see *Tivoli Workload Scheduler: Planning and Installation Guide*.

Quick reference: how to modify log and trace levels

Quick reference information on how to modify log and tracing levels for all components.

Modify log and trace levels for the following components:

Modify Tivoli Workload Scheduler logging level

1. Edit `<TWA_home>/TWS/TWSCCLog.properties`
2. Modify `twc.loggers.msgLogger.level`.

This determines the type of messages that are logged. Change this value to log more or fewer messages, as appropriate, or on request from IBM Software Support. Valid values are:

INFO All log messages are displayed in the log. The default value.

WARNING

All messages except *informational* messages are displayed.

ERROR

Only *error* and *fatal* messages are displayed.

FATAL

Only messages which cause Tivoli Workload Scheduler to stop are displayed.

3. Save the file. The change is immediately effective.

See “Engine log and trace customization” on page 15 for more details.

Modify Tivoli Workload Scheduler tracing level

1. Edit `<TWA_home>/TWS/TWSCCLog.properties`
2. Modify `twc.loggers.trc<component>.level`.

This determines the type of trace messages that are logged. Change this value to trace more or fewer events, as appropriate, or on request from IBM Software Support. Valid values are:

DEBUG_MAX

Maximum tracing. Every trace message in the code is written to the trace logs.

DEBUG_MID

Medium tracing. A medium number of trace messages in the code is written to the trace logs.

DEBUG_MIN

Minimum tracing. A minimum number of trace messages in the code is written to the trace logs.

INFO All *informational*, *warning*, *error* and *critical* trace messages are written to the trace. The default value.

WARNING

All *warning*, *error* and *critical* trace messages are written to the trace.

ERROR

Only *error* and *critical* messages are written to the trace.

CRITICAL

Only messages which cause Tivoli Workload Scheduler to stop are written to the trace.

3. Save the file. The change is immediately effective.

See “Engine log and trace customization” on page 15 for more details.

Modify Dynamic Workload Console tracing level

Follow these steps to activate the Dynamic Workload Console traces at run time:

1. Log in to the Dynamic Workload Console as administrator of the WebSphere Application Server
2. In the Dynamic Workload Console navigation pane select **Settings > Websphere Admin Console**
3. Click Launch **Websphere Admin Console**.
4. In the navigation tree, click **Troubleshooting > Logs and Trace > server name** (for example **tdwcserver**) > **Diagnostic Trace**.
5. Select:

Configuration

If you want to apply the changes to the trace settings after having restarted the server.

Run time

If you want to apply the changes to the trace settings without restarting the server.

6. Click **Change Log Detail Levels** under Additional Properties.
7. Choose the packages for which you want to activate the traces. For the Dynamic Workload Console traces, make this selection:
 - a. Scroll down to **com.ibm.tws.*** and expand the tree
 - b. Click **com.ibm.tws.webui.***
 - c. Either select **All Messages and Traces** or click **Messages and Trace Levels** and choose the trace level you require.

- d. Click **OK** > **Save**.
8. Stop and start the server, if necessary.

Alternatively, you can activate the Dynamic Workload Console traces as follows:

1. Edit the following XML file:

Installed on the WebSphere Application Server:

```
JazzSM_profile_dir/config/cells/JazzSMNode01Cell/nodes/JazzSMNode01/servers/server1/server.xml
```

where, the default value of *JazzSM_profile_dir* is:

On Windows operating systems

```
C:\Program Files\IBM\JazzSM\profile
```

On UNIX operating systems

```
/opt/IBM/JazzSM/profile
```

2. Change the value assigned to the property **startupTraceSpecification** from:
com.ibm.tws.webui.*=info
to:
com.ibm.tws.webui.*=all.
3. Save the changes
4. Stop and start the server.

See: “Activating and deactivating traces in Dynamic Workload Console” on page 31 for more details.

Modify WebSphere® Application Server tracing level

The procedure for changing the trace level on the WebSphere Application Server is as follows:

1. Log on to the computer where Tivoli Workload Scheduler is installed as the following user:

UNIX root

Windows

Any user in the *Administrators* group.

2. Access the directory: <TWA_home>/wastools
3. Run the script:

UNIX

```
./changeTraceProperties.sh -user <TWS_user>  
-password <TWS_user_password>  
-mode <trace_mode>
```

Windows

```
changeTraceProperties.bat -user <TWS_user>  
-password <TWS_user_password>  
-mode <trace_mode>
```

where: <trace_mode> is one of the following:

active_correlation

All communications involving the event correlator are traced.

tws_all_jni

All communications involving the jni code are traced. The jni code

refers to code in shared C libraries invoked from Java™. This option is used by, or under the guidance of, IBM Software Support.

tw_s_all

All Tivoli Workload Scheduler communications are traced.

tw_s_alldefault

Resets the trace level to the default level imposed at installation.

tw_s_cli

All Tivoli Workload Scheduler command line communications are traced.

tw_s_conn

All Tivoli Workload Scheduler connector communications are traced.

tw_s_db

All Tivoli Workload Scheduler database communications are traced.

tw_s_info

Only information messages are traced. The default value.

tw_s_planner

All Tivoli Workload Scheduler planner communications are traced.

tw_s_secjni

All Tivoli Workload Scheduler jni code auditing and security communications are traced. The jni code refers to code in shared C libraries invoked from Java. Only use this option under the guidance of, IBM Software Support.

tw_s_utils

All Tivoli Workload Scheduler utility communications are traced.

tw_s_broker_all

All dynamic workload broker communications are traced.

tw_s_broker_rest

Only the communication between dynamic workload broker and the agents is traced.

tw_s_bridge

Only the messages issued by the workload broker workstation are traced.

4. Stop and restart the application server, as described in the section on starting and stopping the application server in the *Tivoli Workload Scheduler: Administration Guide*.

See “Setting the traces on the application server for the major Tivoli Workload Scheduler processes” on page 37 for more details.

Managing dynamic agent tracing level

To manage traces for the dynamic agent, refer to the following sections:

- “See command usage and verify version” on page 35
- “Enable or disable trace” on page 35
- “Set trace information” on page 35
- “Show trace information” on page 35
- “Collect trace information” on page 36

You can also configure the traces when the agent is not running by editing the [JobManager.Logging] section in the JobManager.ini file as described in Configuring the agent section. This procedure requires that you stop and restart the agent.

Difference between logs and traces

Describes the difference between log and trace messages, and indicates in which languages they are available.

Tivoli Workload Scheduler and the Dynamic Workload Console create both log and trace messages:

Log messages

These are messages that provide you with information, give you warning of potential problems, and inform you of errors. Most log messages are described in *Tivoli Workload Scheduler: Messages*. Log messages are translated into the following languages:

- Chinese - simplified
- Chinese - traditional
- French
- German
- Italian
- Japanese
- Korean
- Portuguese - Brazilian
- Spanish

Messages are written to the log file in the language of the locale set on the computer where they were generated, at the moment when they were generated.

Trace messages

These are messages for IBM Software Support that provide in depth information about Tivoli Workload Scheduler processes. In most cases they are in English. Whereas log messages are written so that you can understand them in relation to the activity you were performing, trace messages might not be. There is no guarantee that you can diagnose any error situations from the information they contain.

The traces are provided at several different levels and in several different forms:

Messages for IBM Software Support

These are similar to log messages, and while not intended for customer use, can be sometimes helpful to experienced customers who know the product well. The information they contain is used by IBM Software Support to understand problems better.

Specific software traces

These are traces written directly by the program code normally indicating the values of variables being used in complex processes. They are not for use by the customer.

Automatic software traces

These are traces issued automatically by the code when it enters and exits code modules. They are not for use by the customer.

The following table gives more detailed information:

Table 2. Difference between logs and traces

Characteristics	Log Messages	Messages for IBM Software Support	Specific software traces	Automatic software traces
Translated	√			
Documented in Information Center	√	Some		
Written to <TWA_home>/TWS/stdlist/logs/	√			
Written to <TWA_home>/TWS/stdlist/traces/	√	√	√	
Logging level, format etc. controlled by TWSCCLog.properties	√	√	√	
Logging level, format etc. controlled by TWSFullTrace				√
Optionally written to memory by TWSFullTrace and written to disc by that utility when requested.	√	√	√	√

If you want to merge the logs and traces controlled by TWSCCLog.properties into one file, set the **localopts** option merge stdlist to yes.

Note: It is also possible to merge these two sets of messages using the correlate logs facility of the Log Analyzer; see “Engine Log Analyzer” on page 18

Tivoli Workload Scheduler logging and tracing using CCLog

Describes the log and trace files created by the CCLog logging engine, and how they are configured.

CCLog is a logging engine that creates log files in a defined structure. It can be used to monitor many products from a variety of software suppliers. The configuration supplied with Tivoli Workload Scheduler uses it uniquely for the processes of Tivoli Workload Scheduler.

The CCLog engine is used wherever any of the following components are installed:

- Master domain manager
- Backup master domain manager
- Fault-tolerant agent

The contents of this section are as follows:

- “Engine log and trace file locations”
- “Engine log and trace file switching” on page 15
- “Engine log and trace customization” on page 15
- “Engine log and trace performance” on page 18
- “Engine Log Analyzer” on page 18

Engine log and trace file locations

Describes where to find the engine log and trace files produced by CCLog.

All log and trace files produced by Tivoli Workload Scheduler are stored in:

```
<TWA_home>/TWS/stdlist/logs/  
<TWA_home>/TWS/stdlist/traces/
```

The files have different names, depending on the settings in the `localopts` file:

merge stdlists = yes

- `<yyyymmdd>_NETMAN.log`
This is the log file for netman.
- `<yyyymmdd>_TWSMERGE.log`
This is the log file for all other processes

merge stdlists = no

`<yyyymmdd>_<process_name>.log`

where `<process_name>` is one of the following:

APPSRVMAN
BATCHMAN
CONNECTR
JOBMAN
JOBMON
MAILMAN
NETMAN
WRITER

Low-level traces, and open source library messages that do not conform to the current standard Tivoli Workload Scheduler message format (for instance, some SSL stdout and stderr messages), are found in the following files:

`<yyyymmdd>/<process_name>`, where `<process_name>` is as above. For more information, see the *Tivoli Workload Scheduler: User's Guide and Reference*.

Note: You can add a local option `restricted stdlists` to your `localopts` file to limit access to the `stdlist` directory on your UNIX workstation. See the *Tivoli Workload Scheduler: Administration Guide* for details.

Engine log and trace file switching

Describes when new log and trace files with the next day's datestamp are created.

The Tivoli Workload Scheduler log files are switched every day, creating new log files with the new datestamp, at the time set in the `startOfDay` global options (`optman`).

Engine log and trace customization

Describes how you can customize the CCLog logging and tracing facility. You can modify the appearance of the log and the logging and tracing levels.

You can customize the information written to the log files by modifying selected parameters in its properties file. The changes you can make affect the format of the log or trace file and the logging level or trace level.

Attention: Do not change any parameters in this file other than those detailed here, otherwise you might compromise the logging facility.

The CCLog properties file is as follows:

```
<TWA_home>/TWS/TWSCCLog.properties
```

where `<TWA_home>` is the directory where Tivoli Workload Scheduler is installed.

Parameters

The parameters that can be modified are as follows:

Logging level

`twc.loggers.msgLogger.level`

This determines the type of messages that are logged. Change this value to log more or fewer messages, as appropriate, or on request from IBM Software Support. Valid values are:

INFO All log messages are displayed in the log. The default value.

WARNING
All messages except *informational* messages are displayed.

ERROR
Only *error* and *fatal* messages are displayed.

FATAL
Only messages which cause Tivoli Workload Scheduler to stop are displayed.

Tracing level

`twc.loggers.trc<component>.level`

This determines the type of trace messages that are logged. Change this value to trace more or fewer events, as appropriate, or on request from IBM Software Support. Valid values are:

DEBUG_MAX
Maximum tracing. Every trace message in the code is written to the trace logs.

DEBUG_MID
Medium tracing. A medium number of trace messages in the code is written to the trace logs.

DEBUG_MIN
Minimum tracing. A minimum number of trace messages in the code is written to the trace logs.

INFO All *informational*, *warning*, *error* and *critical* trace messages are written to the trace. The default value.

WARNING
All *warning*, *error* and *critical* trace messages are written to the trace.

ERROR
Only *error* and *critical* messages are written to the trace.

CRITICAL
Only messages which cause Tivoli Workload Scheduler to stop are written to the trace.

Component names used in the `twc.loggers.trc` property names are for the most part self-explanatory, but the following short explanations might help:

Logger

The main internal component of Tivoli Workload Scheduler that performs the scheduling activities.

Sendevnt

The event processor.

Connectr

The connector.

Log format parameters**fomatters.basicFmt.dateTimeFormat**

This contains a specification of the date and time format used by CCLog when adding the date and time stamp to the message header. The format uses the standard *strftime* format convention, used by many programming libraries. The full format details can be found by searching the Internet, but a synthesis of the commonly used definitions is included in Appendix B, "Date and time format reference - strftime," on page 219.

fomatters.basicFmt.separator

This defaults to the pipe symbol "|", and is used to separate the header of each log message, which contains information such as the date and time stamp and the process that issued the error, from the body, which contains the process-specific information such as the issuing process, the message number and the message text. You can change the separator to another character or characters, or set it to null.

twshnd.logFile.className

This indicates if CCLog uses semaphore memory to write to the log file. The default setting (*ccg_filehandler*) tells CCLog to write each line of a multiline message separately. Each process interleaves each line of its multiline messages with messages from other processes, if necessary, improving performance. While this approach could potentially make the log files more difficult to read, this interleaving only occurs in extreme situations of very high use, for example when many jobs are running concurrently.

The setting *ccg_multiproc_filehandler*, defines that each process completes writing any log message, including multiline messages, before freeing the log file for another process to use. This can have an impact on performance when many processes are running concurrently.

twshnd.loggers.className

This indicates the type of log layout you want to use, determining the number of fields in the log record header. The default setting (*ccg_basiclogger*) tells CCLog to put just the date/time stamp and the process name in the header. The alternative setting is *ccg_pdlogger*, which contains more information in the header, thus reducing the length of the log records available for the message text.

twshnd.loggers.organization

This defaults to *IBM* and is used to differentiate between log entries from applications from different suppliers when the same instance of CCLog is being used by more than one software supplier. Tivoli Workload Scheduler is supplied with a unique

instance, and thus unique log files, so if this value is prefixed to your log messages, you can set the value of this parameter to null to avoid it being displayed.

twsl.loggers.product

This defaults to *TWS* and is used to differentiate when the same log files are used by more than one product. Tivoli Workload Scheduler is supplied with unique log files, so if this value is prefixed to your log messages, you can set the value of this parameter to null to avoid it being displayed.

Other parameters

No other parameters must be modified. To do so risks compromising the logging or tracing facility, or both.

Making changes effective

Making your changes effective depends on the type of change:

Changes to log or trace levels

If you change the `twsl.loggers.msgLogger.level` or the `twsl.loggers.trc<component>.level`, the change is immediately effective after the file has been saved.

All other changes

Restart Tivoli Workload Scheduler to make overall changes effective; restart a process to make process-specific changes effective.

Engine log and trace performance

Describes what impact logging and tracing has on the product's performance.

If you use the default configuration, CCLog does not normally have a significant impact on performance. If you believe that it is impacting performance, check that the default values for the parameters `twshnd.logFile.className` and `twslloggers.className` are as described in "Engine log and trace customization" on page 15, and have not been set to other values.

However, even if the default parameters are in use, you might find that in situations of very heavy workload, such as when you have many jobs running simultaneously on the same workstation, multiline log messages become interleaved with messages from other processes. The length of log messages has been increased to offset this risk, but if you find it becoming a problem, contact IBM Software Support for advice on how to reset the previous settings, which avoided the interleaved messages, but had an impact on performance at busy times.

Engine Log Analyzer

Use Log Analyzer to display log details from the Tivoli Workload Scheduler engine log files, and compare one or more log files. It has facilities to filter log messages by a variety of criteria, reorder log messages by a variety of criteria, and search for specific messages. You can correlate two or more log files from different computers (in different time zones, if required) and select common or corresponding messages. Log Analyzer uses Eclipse technology.

Note: Various websites are indicated in the following procedures. These websites are not owned or controlled by IBM. The following steps were correct at time of

writing, but might be different when you perform them. If one or more of the items discussed below is not available, contact IBM Software Support for assistance.

The information about Log Analyzer is in these sections:

- “Installing Eclipse and the Test and Performance Tools Platform”
- “Installing and configuring the Log Analyzer plug-in” on page 20
- “Upgrading Log Analyzer” on page 20
- “Adding a log file” on page 21
- “Using Log Analyzer” on page 22

Installing Eclipse and the Test and Performance Tools Platform

Eclipse is an open source community whose projects are focused on providing an extensible development platform and application frameworks for building software.

Log Analyzer requires Eclipse, version 3.1, or higher. It is available for the Windows and Linux operating systems (see website for full details). Tivoli Workload Scheduler uses Eclipse version 3.0 as its platform of choice for the Tivoli Information Center. However, *Eclipse, version 3.0 cannot be used for Log Analyzer* because Log Analyzer requires a higher version.

Log Analyzer also requires the *Test and Performance Tools Platform*, version 4.1, or higher.

To install Eclipse and the Test and Performance Tools Platform, follow these steps:

1. Check that you have Java run time environment (JRE) or Java development kit (JDK), version 1.4.2 or higher installed on your machine in order to run Eclipse. If you do not have the appropriate level of JRE or JDK, follow these steps:
 - a. Go to www.java.com
 - b. Download and install Java Standard Edition (Java SE), version 1.4.2, or higher. At time of writing, this could be found by clicking **Free Java Download** on the home page.
 - c. Follow the instructions on the website for downloading and installing J2SE.
2. Go to the Eclipse website at <http://www.eclipse.org/>
3. Click **Downloads**.
4. Under **Third Party Distros**, click **IBM**.
5. In the description of the **Europa testing project bundle**: you should see **Eclipse Test and Performance Tools Platform (TPTP)**. This contains both the prerequisite versions of Eclipse and the Test and Performance Tools Platform. Click **Europa testing project bundle: → Free download**.
6. Save the .zip (Windows) or .gz (UNIX) file containing the Test and Performance Tools files in a temporary directory.
7. Open the .zip or .gz and extract the files to a temporary directory.

Configuring the Log Analyzer memory:

After installing Eclipse you must configure the memory usage for the Tivoli Workload Scheduler plug-in. Do the following:

1. Close Eclipse.
2. Edit the `eclipse.ini` file in the Eclipse install directory.
3. Set the following options:

--launcher.XXMaxPermSize

Set to:

512m

-vmargs

Set to:

-Xms100m

-Xmx512m

When you have finished, your file should look like this:

```
-showsplash  
org.eclipse.platform  
--launcher.XXMaxPermSize  
512m  
-vmargs  
-Xms100m  
-Xmx512m
```

4. Start Eclipse
5. Select **Window → Preferences**
6. Expand the **Java** option
7. Click **Installed JREs**
8. Double-click the **Installed JRE** that you are using (the one in the list that is selected by a check box)
9. In the Edit JRE window, add the following to the field **Default VM Arguments**:
-Xms100m -Xmx512m
10. Close Eclipse.

Eclipse is now ready for use with the Tivoli Workload Scheduler plug-in.

Installing and configuring the Log Analyzer plug-in

What you have installed up to now is generic software for analyzing log files. You now need to install the plug-in that Eclipse uses to read and analyze the specific Tivoli Workload Scheduler log files. To install and configure the Log Analyzer plug-in do the following:

1. Find the Tivoli Workload Scheduler plug-in located on the DVD *IBM Tivoli Workload Scheduler 8.6 Integrations, Multiplatform Multilingual* for your platform, in the following path:

```
TWS_INTEGRATION\integrations\log_analyzer\TWSLogParser.tar
```

This is a compressed archive, which contains just one file:

```
TWSLogParser_8.6.0.jar
```

2. Extract the file into the Eclipse directory, and it is automatically placed in the Eclipse/plugins directory. For example, on Windows, if the location you chose to install Eclipse and the Test and Performance Tools Platform was D:\, you should specify to install the jar file in D:\eclipse

The installation of the Log Analyzer is now complete.

Upgrading Log Analyzer

If you have already installed and used Log Analyzer in a previous release of Tivoli Workload Scheduler you can upgrade the analyzer to be able to use the additional facilities offered in the latest version of Eclipse, details of which can be found on the Eclipse website: <http://www.eclipse.org/>.

To upgrade the Log Analyzer, follow these steps:

1. Delete the existing Eclipse folder and all its plug-ins.
2. Install and configure the new version.
3. Import the log files as described in the following sections.

If you upgrade to this version you should also import the new symptom catalog (formerly called a symptom database), because the format of the catalog has changed (see “Analyzing messages with a symptom catalog” on page 28) for details of the advantages of using the symptom catalog.

Adding a log file

Each log file that you want to look at or analyze must be identified to Log Analyzer, as follows:

1. Run **Eclipse**.
2. From the **File Menu** select **Import**.
3. From the list of import sources, select **Profiling and Logging → Log File**. Click **Next**.
4. On the **Import Log File** panel, select **Add**.
5. On the **Add Log File** panel, select **Tivoli Workload Scheduler stdlist file** from the list of log file types.
6. Click on the **Details** tab of the log file properties:

The screenshot shows the 'Add Log File' dialog box. The 'Log types' section has a filter dropdown set to 'type in a log types filter' and a 'Show categories' checkbox. The list of log types includes 'Common Base Event XML log', 'Java Logging XML log', 'Microsoft Windows Application log', 'Microsoft Windows Security log', 'Microsoft Windows System log', and 'Tivoli Workload Scheduler stdlist file'. The 'Log details' section is active, showing the 'Details' tab. It contains several input fields: 'Absolute path of the log file' with a 'Browse...' button, 'The Tivoli Workload Scheduler workstation from which the log file came (leave as UNKNOWN to get the workstation name from the log file)' with a text field containing 'UNKNOWN', 'The time zone offset for the workstation from GMT' with a text field containing '+01:00', 'The offset in seconds of the log file with respect to other log files already imported' with a text field containing '0', 'Release of Tivoli Workload Scheduler used to generate the log file' with a text field containing '8.5', and 'Tivoli Workload Scheduler CCLOG properties file (leave as UNKNOWN to use the default date format and separator values)' with a text field containing 'UNKNOWN' and a 'Browse...' button. At the bottom are 'OK' and 'Cancel' buttons.

7. Enter or browse for the following information:

Absolute path of the log file

Enter or browse for the absolute path of the log file that you want to load. See Chapter 2, "Logging and tracing," on page 9 for information about the location of log files.

The Tivoli Workload Scheduler workstation name

Leave as "UNKNOWN" and Log Analyzer fills in the information when it loads the file.

Time zone offset for the workstation from GMT

Enter the time zone offset from GMT of the workstation where the log file was recorded, in the format:

±hh:mm

The default is the time zone offset of the workstation where Log Analyzer is being run.

The offset in seconds of the log file with respect to other log files already imported

Enter any additional offset, in seconds, that this log file has from other log files already imported. The default is zero.

Release of Tivoli Workload Scheduler used to generate the log file

Enter the release of Tivoli Workload Scheduler that was running on the workstation when the log file was created. The default is 8.6.

Tivoli Workload Scheduler CCLOG properties file

Enter or browse for the path of the TWSCCLog.properties file (see "Engine log and trace customization" on page 15 for the location). If the log file you want to analyze is not a CCLog file, use the properties file appropriate for the log file, or leave the field as "UNKNOWN" if you want Log Analyzer to use the default values for the date format and field separator values.

8. Click **OK**.
9. Click **Finish** on the Import Log File panel.
10. If the Confirm Perspective Switch window opens, inviting you to switch to the Profiling and Logging Perspective, click **Yes**.

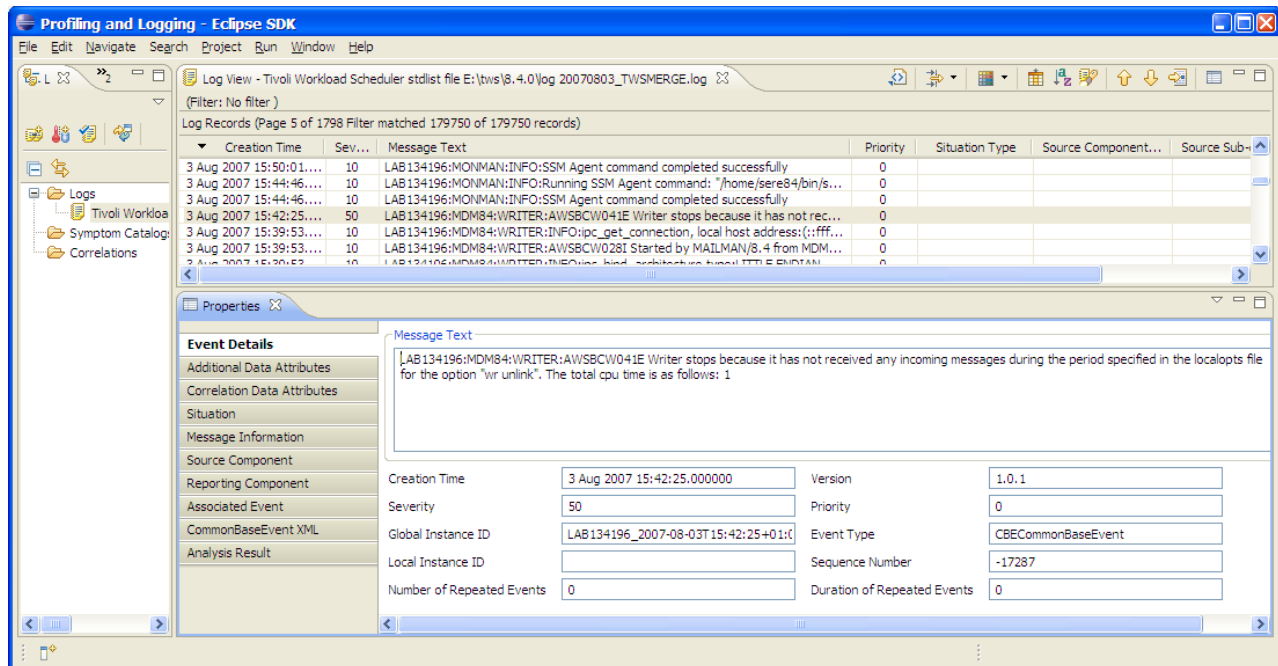
Using Log Analyzer

To use Log Analyzer follow these steps:

1. Run Eclipse.
2. Select a log file that you have already added (see "Adding a log file" on page 21).
3. Use the Log Analyzer options to examine and analyze the data in the file. The available options are the following:
 - "Log Analyzer main window" on page 23
 - "Navigating the log messages" on page 24
 - "Locating a specific message" on page 24
 - "Sorting messages" on page 24
 - "Filtering messages" on page 24
 - "Creating reports" on page 26
 - "Managing the log message properties" on page 26
 - "Comparing log files" on page 28

- “Analyzing messages with a symptom catalog” on page 28

Log Analyzer main window: After you have run **Eclipse** and the Log Analyzer window has opened with a log file already added, you see a window like the following:



The window tabs are as follows:

Log Navigator tab

This is where your log files are listed. Correlations are created by you (see “Comparing log files” on page 28), and you can work with symptom catalogs (see “Analyzing messages with a symptom catalog” on page 28)

Log View tab

The main tab is the Log View tab. This is a list of the records in the log file. An error message with a severity of 50 has been highlighted (severities higher than the standard 10 are highlighted in yellow or red, depending on the severity, but the color disappears when you click on the message to select it).

When a message is highlighted, its details appear in the Properties tab, below. If the Properties tab is not showing, right-click the message you want to examine and select **Properties**.

Above the Log View tab are the icons that you use to perform the functions of Log Analyzer.

Properties tab

This contains several panes of information about the message. Those which contain information with respect to Tivoli Workload Scheduler messages are Event Details, Additional Data Attributes, and CommonBaseEvent XML.

For general help for using Eclipse select **Help → Help Contents**.

For specific help for using Log Analyzer select **Help → Dynamic Help**




Navigating the log messages: To follow the message flow, scroll down the Log Record list. Logs are listed in pages of 50 messages.

The navigation of this list is as follows:

Moving within a page


Use the scroll bars to move up and down within a page. Your keyboard's **PageUp** and **PageDown** keys move the display up and down within a single page.

Moving between pages

To move from one page to the next click the **Page-Down** icon:  icon or the **Page-Up** icon: . Alternatively, you can jump to a particular page by clicking the **Go To Page** icon:  and entering a page number.

Locating a specific message:

To locate a specific message, follow these steps:






1. Click the **Find Log Record** icon: .
2. In the Find Log Record window, click **Add** to define a search expression, by selecting a property and an operator, and entering the value or partial value for the property to search for. Wildcards can be used for the partial value.

For example, selecting the *Message Text* property with "=" (equals), and supplying a value of **AWSBCW041E** creates a search expression that, when you click Find Next, locates the first message containing the string "AWSBCW041E".

These expressions are saved automatically and permanently in the Find Log Record window. On a subsequent visit to this window you can select a search expression you have previously created or add a new one.

Sorting messages:


Messages are presented by default in ascending order of *Creation Time*. If you want to change this order, follow these steps:

1. Click the **Sort Columns...** icon: .
2. Use the central arrow buttons  and  to move selected properties to and from the Properties list and the Selected Properties list.
3. Use the  and  buttons to move properties in the Selected Properties list into the correct sort sequence.
4. Click **OK**. The messages are redisplayed in the selected sequence.

Filtering messages: Many log files are very large, and you might only be interested in a subset of the messages in them. A filter can be applied in Log Analyzer which restricts the messages on display to those that match the filter criteria. You can do the following:

Apply an existing filter

To apply a defined filter, click the arrow beside the **Manage Filters...** icon:

 to choose a filter from those you have already created yourself and the default filters (such as "All error messages"). Filters are not cumulative, so, for example, if you apply a filter for "Error messages", and then apply one that you have created for "All MAILMAN messages", you get a list of "All MAILMAN messages", not "All MAILMAN error messages".

Apply no filter

To stop the effect of the currently applied filter, click the arrow beside the **Manage Filters...** icon and select **No Filter**.

Create a new filter when no filter is in force

If no filter is in force, click the **Manage Filters...** icon to open the Filters panel and create a new filter (see "Adding a new filter" for details on the filter options available).

Create a new filter when another filter is in force

To create a new filter when another filter is in force, click the arrow beside the **Manage Filters...** icon and select the **Manage Filters...** option. From the Add/Edit/Remove Filters window click **New** (see "Adding a new filter" for details on the filter options available).

Edit a filter currently in force

If you have applied a filter and want to edit it, click the **Manage Filters...** icon to open the Filters panel and edit the filter currently in force (see "Adding a new filter" for details on the filter options available).

Edit any other filter

To edit an existing filter, click the arrow beside the **Manage Filters...** icon and select the **Manage Filters...** option. From the Add/Edit/Remove Filters window select a filter to edit, and click **Edit** (see "Adding a new filter" for details on the filter options available).

Delete (remove) a filter

To delete a filter, click the arrow beside the **Manage Filters...** icon and select the **Manage Filters...** option. From the Add/Edit/Remove Filters window click **Remove**.

Adding a new filter:

To add a new filter In the Filters panel, follow this procedure:

1. Give a name to the filter.
2. Decide if you want to set either of the options on the **Standard** tab:

Show events by severity

Set this to select that the filter includes only specific types of message

Show correlated log records only

Select this if you are using a correlation, and want the filter to include only messages that are correlated. See "Comparing log files" on page 28 for more details about correlations.


3. Click the **Advanced** tab.
4. Click **Add** to add a new filter expression. Note that you can make complex filters by creating an unlimited number of filter expressions.
5. On the Add Filter Property window, select a property and an operator, and enter the value or partial value for the property to filter for. Wildcards can be used for the partial value. These expressions are saved automatically and permanently in the Add Filter Property window when you click **OK**.
6. Click **OK** to close the Edit Filter window.
7. If the Add/Edit/Remove Filters window is open, click **OK** to close it.
8. The new filter is applied immediately. If you have a complex filter or many records, you might have to wait for the results to be visible.

For example, creating a filter expression selecting the *Message Text* property with "=" (equals), and supplying a value of **JOBMON**, and then creating a second filter expression selecting the *Creation time* property with ">" (greater than), and

supplying a value of `2008-02-08 21:53:16.38+0100` creates a filter that, when you apply it, displays only messages containing the string "JOBMON" created after the indicated date.

Creating reports:

Reports of selected log details can be created in CSV, HTML, or XML formats, as follows:

1. Use the other facilities described in the above sections to select the messages for which you want to create a report.
2. Ensure that you only have the required properties selected, because the report is created using all of the selected properties. See "Managing the log message properties" for details.
3. Click the **Report ...** icon:  .
4. On the New Report panel select the **Report to be created** (CSV, HTML, or XML).
5. Decide if you want to edit the report after it is created, deselecting the **Open editor** checkbox, if not.
6. Click **Next**.
7. On the Report panel, enter the parent folder for the report, or select one of the listed folders.
8. Supply a file name for the report.
9. If you have selected an HTML Report, you can optionally click **Next** to open a panel where you select which pages of the Log Records view should be included in your report.
10. Click **Finish**. If you selected **Open editor**, the report is displayed as follows:

CSV format

Log Analyzer opens a window in your default application for CSV files (this might be Microsoft Excel, for example, on Windows).

HTML format

A pane is opened at the bottom of the Log Analyzer window.


XML format

A pane is opened at the bottom of the Log Analyzer window.



11. For HTML and XML reports make any changes you require. The pane does not verify the integrity of the HTML or XML after you have edited it, so any changes must be compatible with HTML format or the DTD or schema of the XML file, as appropriate.
12. If you have made any changes, when you click on the **Close** icon you are asked if you want to save the changed file.

Managing the log message properties:

The message properties are not only displayed in the Property and Value pane, but also used for the search, sort, and filter actions. Some of the message properties might not be of interest to you. For example, there is a default property called *priority* that might not interest you. You can hide properties that do not interest you, as follows:

1. Click the **Choose Columns...** icon:  .
2. In the Filter Properties panel are displayed all possible properties that Log Analyzer can manage. Many of them are not properties of Tivoli Workload Scheduler log files, and can be ignored.

Use the central arrow buttons  and  to move selected properties to and from the Properties list and the Selected Properties list.

Use the  and  buttons to move properties in the Selected Properties list into the display order you require (Click the **Sort** buttons on either list to order the properties in alphabetical order).

3. Click OK to finish. Any properties you have selected or deselected are added to or removed from displays and selection panels and drop-downs.



Highlighting messages:

Using the filters described in “Filtering messages” on page 24, you can set a highlight that automatically applies a background color to messages that match the filter in question. For example, by default, messages with a high severity (error messages) display the severity value with a red background; but using this facility you can configure Log Analyzer to display the entire message with a red background.

The following options are available:

Set highlights

Do this as follows:

1. Click the **Highlight Events...** icon: .
2. In the Highlight Events... window select one or more defined filters by clicking their checkbox
3. For each selected filter, click the Color column and then the ellipsis button  that is displayed in the color column.
4. Select, or define and select, the color you require.
5. Click OK to finish. The chosen background color or colors will be applied to the displayed messages.

Note:

- a. You are using the filters only to determine the highlight – whatever filter you might have applied to the messages remains in force, but any displayed messages that match the filters have the chosen background color.
- b. If a message satisfies more than one filter, it is displayed against a black background to warn you of this duplication. To read the black text against the black background, click the message, and the text is displayed in white.

Remove highlights

To remove a highlight, open the Highlight Events window as above and deselect the appropriate filter.

Add new filters

You can add a new filter to the list of defined filters by clicking **New...** (see “Adding a new filter” on page 25 for details on the filter options available)

Edit or delete filters

You can edit or delete a filter from the list of defined filters by clicking the filter name and selecting **Edit...** or **Remove...**, as appropriate (see “Adding a new filter” on page 25 for details on the filter options available)



Show only highlighted events

To show only the highlighted events, click the arrow beside the **Highlight Events...** icon and select **Show only highlighted events**.

Comparing log files:

Two or more log files can be correlated, so that you can compare the messages from each. This might be useful, for example, when comparing a log from the master domain manager with a log from an agent.

To correlate log files take the following steps:

1. Ensure that you have imported the log files you want to correlate.
2. Right-click the **Correlations** folder in the **Log Navigator** tab, and select **New → Log Correlation**.
3. On the New Log Correlation panel give a name to the correlation.
4. Use the central arrow buttons  and  to move selected log files to and from the Available Logs list and the Selected Logs list.
5. Click **Next**.
6. Choose the correlation method:

Tivoli Workload Scheduler Events Correlation

The log files are correlated for matching Tivoli Workload Scheduler events

Tivoli Workload Scheduler Job Execution Correlation

The log files are correlated for matching Tivoli Workload Scheduler jobs

Tivoli Workload Scheduler Linking Correlation

The log files are correlated for corresponding linking and unlinking actions

Time The log files are correlated with respect to time.

Note that the first three correlations can only be performed on files that are in the Tivoli Workload Scheduler stdlist format.

7. Click **Finish**. The chosen log files are correlated.

The correlated log files can now be viewed in one of three ways:

Log View

This is the default. It shows the correlated messages in the first of the chosen log files. Select another log file in the **Log Navigator** pane to see the correlated messages in that file. To return to this view after working with one of the others, right-click on the **Correlation** in the **Log Navigator** pane and select **Open With → Log View**.

Log Interactions

Right-click on the **Correlation** in the **Log Navigator** pane and select **Open With → Log Interactions**. A graphic display shows how the two log files interact.

Log Thread Interactions

Right-click on the **Correlation** in the **Log Navigator** pane and select **Open With → Log Thread Interactions**. A graphic display shows how the two log files interact for individual threads.

Analyzing messages with a symptom catalog: Tivoli Workload Scheduler messages in the log file contain just the message text. To store more information about a message, or to document a course of action in respect of that message, you can create a symptom catalog, recording information in the catalog for any message that could appear in the log.

The symptom catalog is in the form of an xml file. The dtd of the xml file is simple, and can be determined by looking at the symptom catalog supplied with Tivoli Workload Scheduler.

This symptom catalog contains the message help information (explanation, system action, and operator response) for all of the messages that are logged in the Tivoli Workload Scheduler logs (from the *Maestro*, *Unison*, *Netman*, *Cluster*, and *Altinst* catalogs). To determine which messages these are, look at the beginning of each message set described in *Tivoli Workload Scheduler: Messages* – those belonging to the above-mentioned catalogs are indicated. This information is available in English, only. You can use this catalog as it is, modify the catalog, adding information pertinent to your enterprise, or create your own catalog, based on the structure of the example. Log Analyzer supports the contemporaneous presence of more than one catalog, though a message can be analyzed by only one catalog at a time.

Note: Not included are those messages logged in the log files of the application server.

The following sections describe how you do the following:

- “Installing the Tivoli Workload Scheduler symptom catalog”
- “Using the symptom catalog”

Installing the Tivoli Workload Scheduler symptom catalog:

The Tivoli Workload Scheduler symptom catalog is included in the TWSLogParser.tar that you have already installed. However, it needs to be separately imported into Log Analyzer, as follows:


1. Open the TWSPLUGINS/TWSLogParser.tar, described in “Installing and configuring the Log Analyzer plug-in” on page 20 (a zip-like utility can be used).
2. Open the TWSLogParser_8.6.0.jar, contained therein (a zip-like utility can be used).
3. Extract the TWSSymptomDB.symptom into a temporary directory.
4. Start **Eclipse**.
5. From the **File Menu** select **Import**.
6. From the list of import sources, select **Symptom Catalog File** and click **Next**.
7. On the Symptom Catalog File panel, select the **Local Host** radio button.
8. Navigate to and select the TWSSymptomDB.symptom file in the temporary directory created in step 3.
9. Click **Finish** on the Import Symptom Catalog File panel.

The installation of the example symptom catalog is now complete. Use a similar procedure to install your own symptom catalog, should you decide to create one.

Using the symptom catalog:

If you have installed a symptom catalog (see “Installing the Tivoli Workload Scheduler symptom catalog”) take the following steps to see the message help for one or more messages.

1. Select the log message which you require to analyze.
2. Right-click the log message and select **Analyze** to analyze just the selected message or **Analyze All** to analyze all messages in the log file page.
3. The message or messages you have chosen to analyze are listed in the Symptom Analysis Results View.

4. Click a message in this view.
5. Click the **Properties** tab.
6. Under **Other symptom properties**, click the message number in the field **Description**.
7. If the message is present in the symptom catalog, the message number will be highlighted in the TWSSymptomDB.symptom Symptom Definitions view.
8. Expand the selection to show the **Rule** and **Effect** entries.
9. Click **Effect**. In the same panel, under **Symptom effect details**, then **Identification properties**, then **Description**, is displayed the message help.
10. Click the associated ellipsis button  to view a panel showing the Explanation, System Action and Operator Response of the message.

Dynamic Workload Console log and trace files

This section describes the Dynamic Workload Console log and trace files, where to find them, and how to modify log and tracing levels.

Table 3 lists the log and trace files created by the Dynamic Workload Console:

Table 3. Locations of log files and trace files

Path	Files	Content
Dynamic Workload Console installed on the WebSphere Application Server: On Windows: <JazzSM_profile_dir>\logs\server1 On UNIX: <JazzSM_profile_dir>/logs/server1, where, the default value of JazzSM_profile_dir is: /opt/IBM/JazzSM/profile	SystemOut.log, SystemErr.log trace.log	The Dynamic Workload Console run time logs and traces.
On Windows: %TEMP%\TWA\tdwc86 On UNIX: \$TMPDIR/TWA/tdwc86 if set, otherwise /tmp/TWA/tdwc86	tdwinstall.log	The Dynamic Workload Console installation log.
	tdwcuninstall.log	The Dynamic Workload Console uninstall log.
	wsadmin.log	The trace file containing the information about the configuration procedures stored during the installation phase.
	securityConfignnnn.log	The Dynamic Workload Console log file containing the details about the installation errors reported in the tdwinstall.log file. The numeric value <i>nnnn</i> is automatically assigned at installation time. Access the tdwinstall.log file to read the filename of the securityConfignnnn.log file.

Note: For information about the path represented by *JazzSM_profile_dir*, see the *Tivoli Workload Scheduler: Planning and Installation Guide*.

Activating and deactivating traces in Dynamic Workload Console

Describes how to activate or deactivate the Dynamic Workload Console traces.

Activating traces

This task activates Dynamic Workload Console traces.

Follow these steps to activate the Dynamic Workload Console traces at run time:

1. Log in to the Dynamic Workload Console as administrator of the WebSphere Application Server
2. In the Dynamic Workload Console navigation pane select **Settings > Websphere Admin Console**
3. Click Launch **Websphere Admin Console**.
4. In the navigation tree, click **Troubleshooting > Logs and Trace > *server name*** (for example **tdwcserver**) > **Diagnostic Trace**.
5. Select:

Configuration

If you want to apply the changes to the trace settings after having restarted the server.

Run time

If you want to apply the changes to the trace settings without restarting the server.

6. Click **Change Log Detail Levels** under Additional Properties.
7. Choose the packages for which you want to activate the traces. For the Dynamic Workload Console traces, make this selection:
 - a. Scroll down to **com.ibm.tws.*** and expand the tree
 - b. Click **com.ibm.tws.webui.***
 - c. Either select **All Messages and Traces** or click **Messages and Trace Levels** and choose the trace level you require.
 - d. Click **OK > Save**.
8. Stop and start the server, if necessary.

Alternatively, you can activate the Dynamic Workload Console traces as follows:

1. Edit the following XML file:

Installed on the WebSphere Application Server:

`JazzSM_profile_dir/config/cells/JazzSMNode01Cell/nodes/JazzSMNode01/servers/server1/server.xml`

where, the default value of `JazzSM_profile_dir` is:

On Windows operating systems

`C:\Program Files\IBM\JazzSM\profile`

On UNIX operating systems

`/opt/IBM/JazzSM/profile`

2. Change the value assigned to the property **startupTraceSpecification** from:
`com.ibm.tws.webui.*=info`
to:
`com.ibm.tws.webui.*=all`.
3. Save the changes

4. Stop and start the server.

When you enable tracing at run time the traces are stored in the following file:

Installed on the WebSphere Application Server:

`<JazzSM_profile_dir>/logs/twaserver<n>/trace.log`, (where `<n>` is null, 1, 2, and so on)

Deactivating traces

This task deactivates Dynamic Workload Console traces.

Follow the instructions for activating traces (see “Activating traces” on page 31), with these differences:

Deactivating traces using the Integrated Solutions Console

When you have selected `com.ibm.tws.webui.*`, select **Messages Only**.

Deactivating traces by editing the startupTraceSpecification configuration

Change the value assigned to the property `startupTraceSpecification` from

`com.ibm.tws.webui.*=all.to`

`com.ibm.tws.webui.*=info`

Dynamic workload scheduling log and trace files

The logs and traces produced by the dynamic workload scheduling processes are in most part included in the log and trace files of the Tivoli Workload Scheduler master domain manager. In addition, the files listed in Table 4 also contain log and trace material from these processes.

Table 4. Locations of log and trace files

Component	Path	Trace files	Log files	Content
Tivoli Workload Scheduler master domain manager	<code><WAS_profile_path> /logs/twaserverN</code> where <i>N</i> is the number of the TWA instance and the default path for <code><WAS_profile_path></code> is <code>TWA_home/WAS/TWSpfile</code>	<code>native_stderr.log</code> <code>native_stdout.log</code> <code>serverStatus.log</code> <code>startServer.log</code> <code>stopServer.log</code> <code>SystemErr.log</code> <code>trace.log</code>	<code>SystemOut.log</code>	Additional log files used by dynamic workload scheduling
Tivoli Workload Scheduler agent	<code>TWA_home /TWS/stdlist/JM</code>	<code>JobManager_trace.log</code> <code>ita_trace.log</code>	<code>JobManager_message.log</code> <code>ita_message.log</code>	Log and trace files
	<code>TWA_home /TWS/stdlist/JM/</code> <code>JOBMANAGER-FFDC/yy-mm-dd/</code>		<code>JobManager_message.log</code>	Processing error log file
Job Brokering Definition Console	<code>user's home directory/jd_workspace/</code> <code>.metadata/tivoli/JBDC/logs</code>	<code>trace.log</code>	<code>msg.log</code> , <code>msg_cbe.log</code>	Trace files
	<code>\$TEMP/TWA/jbdc851</code>	<code>trace_installation.log</code> <code>trace_installation_xml.log</code>	<code>msg_installation.log</code>	Installation log and trace files

Activating logs for Job Brokering Definition Console

By default, logging is disabled. To generate log files, you must enable tracing in the **Preferences** dialog box.

To enable logging, perform the following steps:

1. Select **Preferences** in the Windows menu. The **Preferences** dialog box is displayed.

2. Optionally, specify a path and name for the log file in the **Log file directory** field.
3. Select the **Enable logging to console** check box.

The logs are saved in the directory indicated in Table 4 on page 32.

Dynamic agent log and trace files

Describes how to collect log and trace files for the agent.

You can collect log and trace files for the agent, by performing the following actions:

When the agent is running:

By using the **twstrace** command as described in “twstrace command” on page 34.

When the agent is stopped:

By configuring the [JobManager.Logging] section in the JobManager.ini file as described in Configuring log message properties [JobManager.Logging.cclg]. This procedure requires that you stop and restart the dynamic agent.

The log messages are written in the following file:

On Windows operating systems:

<TWA_home>\TWS\stdlist\JM\JobManager_message.log

On UNIX and Linux operating systems:

<TWA_home>/TWS/stdlist/JM/JobManager_message.log

The trace messages are written in the following file:

On Windows operating systems:

- <TWA_home>\TWS\stdlist\JM\ITA_trace.log
- <TWA_home>\TWS\stdlist\JM\JobManager_trace.log
- <TWA_home>\TWS\JavaExt\logs\javaExecutor0.log

On UNIX and Linux operating systems:

- <TWA_home>/TWS/stdlist/JM/ITA_trace.log
- <TWA_home>/TWS/stdlist/JM/JobManager_trace.log
- <TWA_home>/TWS/JavaExt/logs/javaExecutor0.log

Logging information about job types with advanced options

You can use the logging.properties file to configure the logging process for job types with advanced options, with the exception of the Executable and Access Method job types.

The logging.properties file is located on the Tivoli Workload Scheduler for z/OS Agent, under *TWA_home*/TWS/JavaExt/cfg/logging.properties.

Following installation, this file is as follows:

```
# Specify the handlers to create in the root logger
# (all loggers are children of the root logger)
# The following creates two handlers
handlers = java.util.logging.ConsoleHandler,
           java.util.logging.FileHandler

# Set the default logging level for the root logger
.level = INFO
```

```
# Set the default logging level for new ConsoleHandler instances
java.util.logging.ConsoleHandler.level = INFO

# Set the default logging level for new FileHandler instances
java.util.logging.FileHandler.level
    = ALL
java.util.logging.FileHandler.pattern
    = C:\TWA_home\TWS\JavaExt\logs\javaExecutor%g.log
java.util.logging.FileHandler.limit
    = 1000000
java.util.logging.FileHandler.count
    = 10

# Set the default formatter for new ConsoleHandler instances
java.util.logging.ConsoleHandler.formatter =
    java.util.logging.SimpleFormatter
java.util.logging.FileHandler.formatter =
    java.util.logging.SimpleFormatter

# Set the default logging level for the logger named com.mycompany
com.ibm.scheduling = INFO
```

You can customize:

- The logging level (from INFO to WARNING, ERROR, or ALL) in the following keywords:
 - .level** Defines the logging level for the internal logger.
 - com.ibm.scheduling**
 - Defines the logging level for the job types with advanced options. To log information about job types with advanced options, set this keyword to ALL.
- The path where the logs are written, specified by the following keyword:
 - java.util.logging.FileHandler.pattern**

Configuring trace properties when the agent is running

Use the **twstrace** command to set the trace on the agent when it is running.

Using the **twstrace** command, you can perform the following actions on the agent when it is running:

- “See command usage and verify version” on page 35.
- “Enable or disable trace” on page 35.
- Set the traces to a specific level, specify the number of trace files you want to create, and the maximum size of each trace file. See “Set trace information” on page 35.
- “Show trace information” on page 35.
- Collect trace files, message files, and configuration files in a zip file. See “Collect trace information” on page 36.

You can also configure the traces when the agent is not running by editing the [JobManager.Logging] section in the JobManager.ini file as described in Configuring the agent section. This procedure requires that you stop and restart the agent.

twstrace command

Use the **twstrace** command to configure traces, and collect logs, traces, and configuration files (ita.ini and jobManager.ini) for agents. You collect all the information in a zip file when it is running without stopping and restarting it.

See command usage and verify version

To see the command usage and options, use the following syntax.

Syntax

```
twstrace -u | -v
```

Parameters

- u** Shows the command usage.
- v** Shows the command version.

Enable or disable trace

To sets the trace to the maximum level or disable them, use the following syntax.

Syntax

```
twstrace -enable | -disable
```

Parameters

- enable**
Sets the trace to the maximum level. The maximum level is **3000**.
- disable**
Disables the traces on the agent.

Set trace information

To set the trace to a specific level, specify the number of trace files you want to create, and the maximum size the trace files can reach, use the following syntax.

Syntax

```
twstrace [ -level <level_number> ] [ -maxFiles <files_number> ] [ -maxFileBytes <bytes_number> ]
```

Parameters

- level <level_number>**
Set the trace level. Specify a value in the range from 1000 to 3000.
- maxFiles <files_number>**
Specify the number of trace files you want to create.
- maxFileBytes <bytes_number>**
Set the maximum size in bytes that the trace files can reach. The default is **1024000** bytes.

Show trace information

To display the current trace level, the number of trace files, and the maximum size the trace files can reach, use the following syntax.

Syntax

twstrace -level | -maxFiles | -maxFileBytes

Parameters

-level

See the trace level you set.

-maxFiles

See the number of trace files you create.

-maxFileBytes

See the maximum size you set for each trace file

Sample

The example shows the information you receive when you run the following command:

```
twstrace -level -maxFiles -maxFileBytes
```

```
AWSITA176I The trace properties are: level="1000",  
max files="3", file size="1024000".
```

Collect trace information

To collect the trace files, the message files, and the configuration files in a zip file, use the following syntax.

Syntax

twstrace -getLogs [-zipFile <zip_file_name>] [-host <host_name>] [-protocol {http | https}] [-port <port_number>] [-iniFile <ini_file_name>]

Parameters

-zipFile <zip_file_name>

Specify the name of the zip file that contains all the information, that is logs, traces, and configuration files (ita.ini and jobManager.ini) for the agent. The default is **logs.zip**.

-host <host_name>

Specify the host name or the IP address of the agent for which you want to collect the trace. The default is **localhost**.

-protocol http|https

Specify the protocol of the agent for which you are collecting the trace. The default is the protocol specified in the **.ini** file of the agent.

-port <port_number>

Specify the port of the agent. The default is the port number of the agent where you are running the command line.

-iniFile <ini_file_name>

Specify the name of the **.ini** file that contains the SSL configuration of the agent for which you want to collect the traces. If you are collecting the traces for a remote agent for which you customized the security certificates, you must import the certificate on the local agent and specify the name of the **.ini** file that contains this configuration. To do this, perform the following actions:

1. Extract the certificate from the keystore of the remote agent.

2. Import the certificate in a local agent keystore. You can create an ad hoc keystore whose name must be **TWSCClientKeyStore.kdb**.
3. Create an **.ini** file in which you specify:
 - 0 in the **tcp_port** property as follows:
tcp_port=0
 - The port of the remote agent in the **ssl_port** property as follows:
ssl_port=<ssl_port>
 - The path to the keystore you created in Step 2 in the **key_repository_path** property as follows:
key_repository_path=<local_agent_keystore_path>

Log and trace files for the application server

The log and trace files for the WebSphere Application Server can be found in:

On UNIX:

<WAS_profile_path>/logs/server1/SystemOut.log

<WAS_profile_path>/logs/server1/trace.log

On Windows:

<WAS_profile_path>\logs\server1\SystemOut.log

<WAS_profile_path>\logs\server1\trace.log

where *WAS_profile_path* corresponds to the WebSphere Application Server profile path you specified when you installed one of the following components: master domain manager, backup master domain manager, dynamic domain manager, backup dynamic domain managers. The default path is: *TWA_home*/WAS/TWSPprofile.

Setting the traces on the application server for the major Tivoli Workload Scheduler processes

The application server handles all communications between the Tivoli Workload Scheduler processes. The trace for these communications is set to "twc_info" by default (information messages only). The application server can be set to trace "all" communications, either for the whole product or for these specific groups of processes:

- Command line
- Connector
- Database
- Planner
- Utilities
- Dynamic workload broker

Significant impact on performance: Activating traces for the WebSphere Application Server leads to a significant impact on performance, especially if you set the tracing to "all". Thus you are strongly advised to identify the process group where the problem that you want to trace is occurring, and only set the trace to that group.

The procedure for changing the trace level on the WebSphere Application Server is as follows:

1. Log on to the computer where Tivoli Workload Scheduler is installed as the following user:

UNIX root

Windows

Any user in the *Administrators* group.

2. Access the directory: <TWA_home>/wastools
3. Run the script:

UNIX

```
./changeTraceProperties.sh -user <TWS_user>  
                           -password <TWS_user_password>  
                           -mode <trace_mode>
```

Windows

```
changeTraceProperties.bat -user <TWS_user>  
                        -password <TWS_user_password>  
                        -mode <trace_mode>
```

where: <trace_mode> is one of the following:

active_correlation

All communications involving the event correlator are traced.

twc_all_jni

All communications involving the jni code are traced. The jni code refers to code in shared C libraries invoked from Java. This option is used by, or under the guidance of, IBM Software Support.

twc_all

All Tivoli Workload Scheduler communications are traced.

twc_alldefault

Resets the trace level to the default level imposed at installation.

twc_cli

All Tivoli Workload Scheduler command line communications are traced.

twc_conn

All Tivoli Workload Scheduler connector communications are traced.

twc_db

All Tivoli Workload Scheduler database communications are traced.

twc_info

Only information messages are traced. The default value.

twc_planner

All Tivoli Workload Scheduler planner communications are traced.

twc_secjni

All Tivoli Workload Scheduler jni code auditing and security communications are traced. The jni code refers to code in shared C libraries invoked from Java. Only use this option under the guidance of, IBM Software Support.

twc_utils

All Tivoli Workload Scheduler utility communications are traced.

twc_broker_all

All dynamic workload broker communications are traced.

twc_broker_rest

Only the communication between dynamic workload broker and the agents is traced.

twbridge

Only the messages issued by the workload broker workstation are traced.

4. Stop and restart the application server, as described in the section on starting and stopping the application server in the *Tivoli Workload Scheduler: Administration Guide*.

To reset the traces to the default value, either run the above procedure with `trace_mode` as *twinfo*, or just stop and start the server, as follows:

1. Log on to the computer where Tivoli Workload Scheduler is installed as the following user:

UNIX root

Windows

Any user in the *Administrators* group.

2. Access the directory: `<TWA_home>/wastools`
3. Stop and restart the application server as described in the section on starting and stopping the application server in the *Tivoli Workload Scheduler: Administration Guide*.

Log files for the command line client

The command line client writes its logs in the following files:

UNIX `<command line client install directory>/stdlist/yyyy.mm.dd/
<TWS_user>`

Windows

`<command line client install directory>\stdlist\yyyy.mm.dd\
<TWS_user>`

For example, a log file created on UNIX on December 1, 2008 for the user *myUserID* where the command line client was installed in the default directory is called:

`/opt/ibm/TWS/CLI/stdlist/2008.12.01/myUserID`

Chapter 3. Capturing data in the event of problems

Describes the facilities available for data capture in the event of problems occurring. It provides full details of the Data capture utility and the provisions for first failure data capture.

In the event of any problems occurring while you are using Tivoli Workload Scheduler, you might be asked by the IBM Support Center to supply information about your system that might throw a light on why the problem occurred. The following are available:

- A general data capture utility command that extracts information about Tivoli Workload Scheduler and related workstations; see “Data capture utility.”
- A first failure data capture (ffdc) facility built into **batchman** and **mailman** that automatically runs the data capture utility when failures occur in **jobman**, **mailman**, or **batchman**; see “First failure data capture (ffdc)” on page 50.

Data capture utility

The data capture utility is a script named `twc_inst_pull_info` which extracts information about a product instance of Tivoli Workload Scheduler.

This script collects information that IBM Software Support can use to diagnose a problem. The data capture utility runs on all the supported operating systems.

The data capture utility script is located in the `<TWA_home>/TWS/bin` directory and can be run from the UNIX or DOS prompt on the master domain manager, the backup master domain manager, or a standard or fault-tolerant agent.

When to run the utility

Describes the circumstances in which you would use the data capture utility.

Use the data capture utility in these circumstances:

- A Tivoli Workload Scheduler process has failed, but the automatic ffdc facility has not detected the failure and run the script for you (see “First failure data capture (ffdc)” on page 50)
- Tivoli Workload Scheduler is very slow or is behaving in any other abnormal way
- You are requested to do so by IBM Software Support

Using the utility when you need to switch to the backup master domain manager

If the master domain manager fails you might decide that you want to switch to the backup master domain manager to keep your scheduling activities running. If you also want to run the data capture utility you have two choices:

Data capture first

Run the data capture utility first to ensure that the information extracted is as fresh as possible. Then run **switchmgr**.

To reduce the time between the failure event and the running of **switchmgr**, run the data capture utility without dumping the DB2®

database, then run it again on what is now the backup master domain manager as soon as **switchmgr** has completed, and this time dump the DB2 database.

Switchmgr first

In an emergency situation, where you must continue scheduling activities, run **switchmgr** immediately and then run the data capture utility on both the new master domain manager and the new backup master domain manager as soon as **switchmgr** has completed.

Prerequisites

Describes the prerequisites for running the `twc_inst_pull_info` data capture utility.

The following are the prerequisites for running the data capture utility:

Where the utility can be run

The utility can be run on the master domain manager, the backup master domain manager or a standard or fault-tolerant agent.

Who can run it

The utility must be run by one of the following users:

- Any Tivoli Workload Scheduler user
- Root (recommended on UNIX or Linux systems)
- Administrator (on Windows systems)

To determine the best user to run the script, consider the following:

Troubleshooting any type of problem

- On UNIX operating systems the user running the script must have read access to the `/etc` and `/etc/TWS` directories and read access to the `/etc/TWS/TWSRegistry.dat` file

Troubleshooting installation problems

- On UNIX operating systems, run the script as root to ensure to gather all installation information.

Troubleshooting problems when the product is running

- The script will only extract database object descriptions to which the user running it has EXTRACT permission in the Security file. The `<TWS_User>` (the user who performed the installation) normally has full access to all database objects, so this is the best user to run the script.
- The Tivoli Workload Scheduler instance must have a Symphony file otherwise some information will not be extracted.

Other prerequisites

The facility to dump the database is only available for DB2 databases.

Command and parameters

Describes the command syntax and parameters of the data capture utility.

Command syntax

Run the data capture utility with the following command:

```
twc_inst_pull_info.sh -u
```

```

twins_inst_pull_info.sh
  -twinsuser <userid>
  -log_dir_base <path>
  [-run_db2_module <y/n>]
  [-extract_db_defs <y/n>]
  [-date <yyyymmdd>]

```

This is the syntax for UNIX operating systems; on Windows use

twins_inst_pull_info.cmd

Parameters

-twinsuser

The Tivoli Workload Scheduler user that you specify when you install the Tivoli Workload Scheduler. This user must exist in the /etc/TWS/TWSregistry.dat file if the Tivoli Workload Scheduler instance already exists. This parameter is mandatory.

-log_dir_base

The base directory location where the collected data is stored. The user must have write access to the specified directory. This parameter is mandatory.

-run_db2_module

Only applicable if you are using a DB2 database. Identifies if DB2 related data is to be extracted. This operation might take some time. Valid values are y or n. Set to y if you want to collect DB2 related data. This parameter is optional. The default is n.

-extract_db_defs

Only applicable on the master domain manager. Identifies if database definitions are extracted. Valid values are y or n. This parameter is optional. The default is y.

The Tivoli Workload Scheduler Security access permission (EXTRACT) for the user running the script determines which database objects can be extracted. If the user (including root or Windows Administrator) running the script does not exist in the Tivoli Workload Scheduler Security files, then no database data is extracted.

-date

Used as the base date for collected data logs. If not specified, the script uses the current date by default. Run the data capture utility as soon as a problem occurs, to collect the data specific to the date and time of the problem. Thus, if the problem occurs on the current date, this option is not required. If the problem occurred earlier, then the date on which the problem occurred must be specified in the yyyymmdd format. Either the current date or the specified date is used to identify which files and logs are extracted. This parameter is optional.

-u Displays the usage of the command.

Tasks

Describes the tasks performed by the data capture utility.

Check that the user exists

The script verifies if the specified user exists in the TWSRegistry.dat file. If it does, the <TWS_HOME> directory used for data collection is extracted from the TWSRegistry.dat file. (UNIX only) If the specified user does not exist, the script verifies if the user exists in the /etc/passwd file. If no user exists, the script terminates.

Check the user permissions

The commands that are used during the data collection try to retain the original ownership of the files; when the script is run on Solaris platforms, the ownership of the files might change. If the script is run by a IBM Tivoli Workload Scheduler user (for example, not the root user) the script collects the available instance data.

Note:

Some Windows security policies can affect which data is extracted.

Create the directories in which to store the collected data

The script first creates the <log_dir_base> directory, where <log_dir_base> is the value provided for the -log_dir_base option. Within the <log_dir_base> directory, the script creates the tws_info directory and its subdirectories TWS_YYYYMMDD_HHMMSS, where YYYY=year, mm=month, dd=day, hh=hour, mm=minute and ss=seconds.

Collect data

The script collects system and product-specific data, creating a structure of subdirectories as described in “Data structure” on page 47.

Create the TAR file

UNIX The script creates the TAR file TWS_YYYYMMDD_HHMMSS.tar and compresses it to TWS_YYYYMMDD_HHMMSS.tar.Z, or if the operating system is Linux_i386, TWS_YYYYMMDD_HHMMSS.tar.gz.

Windows

On Windows operating systems there is no built-in zip or tar program, so the script does not create a tar or zip. If you intend to send the data to IBM Software Support you should use your own zip utility to create the compressed archive.

Data collection

Describes the data collected by the data capture utility.

System-specific data

For system-specific data, the script performs the following operations:

- Extracts local CPU node information
- Extracts the environment for the current IBM Tivoli Workload Scheduler instance
- Extracts nslookup information for local CPU
- Extracts netstat information for local CPU
- Extracts Services information
- Extracts the current running processes for the current Tivoli Workload Scheduler user
- Extracts the current available disk space for %TWS_HOME%
- Extracts the current available disk space for the tmp directory
- (UNIX only) Extracts the current system disk space
- (UNIX only) Extracts the current disk space of root filesystem
- (Solaris 10.x or above) Extracts zonecfg information
- (AIX® only) Copies netsvc.conf
- (UNIX only, except AIX) Copies thensswitch.* files

- Copies the host and services files

Tivoli Workload Scheduler-specific data

For Tivoli Workload Scheduler-specific data, the script performs the following operations:

Collects Tivoli Workload Scheduler messages, as follows:

- Generates a list of the .msg files
- Extracts a list of the files in the %TWS_HOME%\ftbox directory

Collects Tivoli Workload Scheduler information, as follows:

- Extracts information about the Tivoli Workload Scheduler instance installation
- Extracts the Tivoli Workload Scheduler Security file
- Extracts a list of the Tivoli Workload Scheduler binaries
- Extracts a list of the files in the %TWS_HOME% directory
- Extracts a list of the files in the %TWS_HOME%\mozart directory
- Extracts a list of the files in the %TWS_HOME%\pids directory
- Extracts a list of the files in the %TWS_HOME%\network directory
- Extracts a list of the files in the %TWS_HOME%\audit\database directory
- Extracts a list of the files in the %TWS_HOME%\audit\plan directory
- Extracts the database definitions to flatfiles
- (UNIX only) Extracts the optman output
- (UNIX only) Extracts planman "showinfo" output
- (UNIX only) Extracts the list of the %TWS_HOME%\trace directory
- Copies jobmanrc.cmd and jobmanrc (if it exists)
- Copies the schedlog files of the previous day (the option -date is not used)
- Copies the schedlog files of the day on which the problem occurred, day - 1 and day + 1 (the option -date is used)
- Copies a list of the files in %TWS_HOME%\audit\database\\${today}
- Copies a list of the files in %TWS_HOME%\audit\database\\${yesterday}
- Copies a list of the files in %TWS_HOME%\audit\plan\\${today}
- Copies a list of the files in %TWS_HOME%\audit\plan\\${yesterday}
- Copies the BmEvents.conf file and the event log (if %TWS_HOME%\BmEvents.conf exists)
- Copies the content of the BmEvents log file (if %TWS_HOME%\BmEvents.conf exists)
- Copies the TWSRegistry.dat file
- Copies the content of the %TWS_HOME%\version directory
- Copies the files of the local workstation (the master domain manager and the backup master domain manager are also workstations on which jobs can be scheduled)
- (Windows only) If the z/OS connector is installed locally, copies the TWSZOSConnRegistry.dat file

Collects Tivoli Workload Scheduler logs, as follows:

- Copies the TWSUser BATCHUP and NETMAN stdlist files for current and previous date

- Copies the TWSMERGE and NETMAN log files from the stdlist\logs directory for current and previous date
- Copies the TWSMERGE BATCHUP and NETMAN stdlist files from the stdlist\traces directory for current and previous date

Collects Tivoli Workload Scheduler files, as follows:

- If the dynamic agent is installed, extracts a list of the files in the %TWS_HOME%\ITA directory
- If the dynamic agent is installed, when twstrace -getLogs is submitted to collect trace files, message files, and configuration files, the result is returned when the data capture utility script is run, tws_inst_pull_info.sh/tws_inst_pull_info.cmd.
- Extracts a list of the files in the %TWS_HOME%\stdlist\JM directory
- Extracts a list of the files in the %TWS_HOME%\jmJobTableDir directory
- If the dynamic agent is installed, copies the *.ini and *.log files in %TWS_HOME%\ITA and %TWS_HOME%\ITA\bin
- If the dynamic agent is installed, copies the *.out files in %TWS_HOME%\ITA and %TWS_HOME%\ITA\bin
- Copies all the files in the %TWS_HOME%\stdlist\JM directory
- Copies all the files in %TWS_HOME%\jmJobTableDir

Collect xtrace information from Tivoli Workload Scheduler processes as follows:

- Generates snapshot files for Tivoli Workload Scheduler processes in raw format
- Generates snapshot files in XML format from the raw format

If Tivoli Workload Scheduler for Applications is installed on the workstation, collects data on the methods, as follows:

- Copies the content of the %TWS_HOME%\methods directory (if it exists)
- (Windows only) Collects information about the Peoplesoft method
- Collects information about the r3batch method
- (UNIX only) Collects the r3batch picklist results

WebSphere-specific data

For WebSphere-specific data, the script performs the following operations:

- (Windows only) Extracts the list of WebSphere logs
- Extracts a list of the <WAS_HOME>/profiles
- Extracts a list of the Tivoli Workload Scheduler server files specific to WebSphere
- Copies the WebShpere logs
- Copies the Tivoli Workload Scheduler specific WebSphere logs
- Copies all the files from %WAS_PROFILE%.deleted (if it exists)
- Copies security.xml of defaultnode
- Copies all the Tivoli Workload Scheduler server files specific to WebSphere
- Collects the javacore*.txt files from the path <WAS_HOME>/profiles/TIPProfile when the data capture utility script is run, tws_inst_pull_info.sh/tws_inst_pull_info.cmd.
- (On UNIX for root user only) Collects the data source properties
- (On UNIX for root user only) Collects the host properties
- (On UNIX for root user only) Collects the security properties

DB2-specific data

For DB2-specific data, the script performs the following operation (if the database is running on other supported database software, no data is collected):

- Collects the DB2 data using the DB2Support tool

Data structure

Describes the data structure created by the data capture utility to contain the extracted data.

Table 5. Collected data structure on UNIX

Gathered data directory structure	TWS filesystem or command	Files and listings
<root_dir>	General collector output	datagather_summary.log,TWS_<today>_files.txt NODE_<hostname>_TWSuser_<twuser>_ Base_Date_<yyyymmdd>.README
<root_dir>/db2_oracle_info	\${db2user_home}/sqllib/db2dump "db2support -d <db2db>" output	db2diag.log db2support.zip
<root_dir>/system_info	"uname -a" output "env" output "nslookup \${local_cpu}" output "netstat -a", "netstat -rn" output "ps -ef grep \${twuser}" output "df -k" output "df -k /" output "df -k \${TWS_HOME}" output "df -k \${TMP_DIR}" output /etc "zonecfg list" output	cpu_node_info.txt instance_env_info.txt cpu_nslookup_info.txt cpu_netstat_info.txt ps_ef_listing.txt system_disk_available.txt root_disk_available.txt tws_home_disk_available.txt tmp_disk_available.txt hosts, services, netsvc.conf (AIX only), nsswitch.* (UNIX, except AIX) zonecfg.txt (Solaris 10.x or higher)
<root_dir>/tws_<version>_install	TWS install, upgrade log files from /tmp/TWA/tws<version> directory	.*
<root_dir>/tws_info	\${TWS_HOME}	Symphony, Sinfonia, StartUp, Jnext*, prodsked, Symnew, Jobtable, localopts, Security_file.txt (output from dumpsec), jobmanrc.txt,.jobmanrc.txt, twshome_files_ list.txt
	\${TWS_HOME}/schedlog	M\${today}*, M\${tomorrow}*, M\${yesterday}* (-date option used)M\${yesterday}* (-date option not used)
	\${TWS_HOME}/mozart \${TWS_HOME}/bin/* \${TWS_HOME}/ftbox \${TWS_HOME}/pids \${TWS_HOME}/network \${TWS_HOME}/audit/database \${TWS_HOME}/audit/database/\${today} \${TWS_HOME}/audit/database/ \${yesterday} \${TWS_HOME}/audit/plan \${TWS_HOME}/audit/plan/\${today} \${TWS_HOME}/audit/plan/\${yesterday} \${TWS_HOME}/BmEvents.conf \${TWS_HOME}/BmE* Composer output on master \${TWS_REGISTRY_PATH} \${TWS_HOME}/version \${TWS_HOME}/bin/optman \${TWS_HOME}/bin/planman "showinfo" \${TWS_HOME}/trace	globalopts, mozart_dir_list.txt tws_binary_list.txt ftbox_dir_list.txt pids_dir_list.txt network_dir_list.txt audit_database_dir_list.txt audit_database_\${today} audit_database_\${yesterday} audit_plan_dir_list.txt audit_plan_\${today} audit_plan_\${yesterday} BmEvents.conf BmEvents_event_log.txt job_defs, sched_defs, cpu_defs, calendar_defs, parms_defs, resource_defs, prompt_defs, user_defs TWSRegistry.dat .* optman_ls_info.txt planman_showinfo.txt trace_dir_image_existing_snap.txt
<root_dir>/tws_ita_files	\${TWS_HOME}/ITA	*.out, ita_dir_list.txt
<root_dir>/tws_ita_bin_files	\${TWS_HOME}/ITA/bin	*.ini, *.log, ita_bin_dir_list.txt

Table 5. Collected data structure on UNIX (continued)

Gathered data directory structure	TWS filesystem or command	Files and listings
<root_dir>/twc_jobmgr_ffdc_files	N/A	--
<root_dir>/twc_jobmgr_ffdc_files/<date>	\${TWS_HOME}/stdlist/JM/JOBMANAGER-FFDC/*	*.*
<root_dir>/twc_jobmgr_files	\${TWS_HOME}/stdlist/JM	*.*, jobmanager_dir_list.txt
<root_dir>/twc_jobstore_files	\${TWS_HOME}/jmJobTableDir/*	*.*, jobstore_dir_list.txt
<root_dir>/twc_logs	N/A	—
<root_dir>/twc_logs/stdlist	N/A	—
<root_dir>/twc_logs/stdlist/<date>/<date>	\${TWS_HOME}/stdlist/<date>	twcuser and netman files from date and date-1
<root_dir>/twc_logs/stdlist/<date>/logs	\${TWS_HOME}/stdlist/logs	twcmerge and netman logs from date and date-1
<root_dir>/twc_logs/stdlist/<date>/traces	\${TWS_HOME}/stdlist/traces	twcmerge and netman traces from date and date-1
<root_dir>/twc_methods	\${TWS_HOME}/methods ./r3batch -v ./r3batch -t PL -c <cpu> -l * -j * -- "-debug -trace"	*.*, methods_dir_list.txt <cpu>_r3batch_ver.txt, r3batch version output <cpu>_r3_batch_info.txt, picklist of scheduled jobs on SAP
<root_dir>/twc_msg_files	\${TWS_HOME} \${TWS_HOME}/pobox	*.msg, msg_file_listing.txt *.msg
<root_dir>/twc_xtrace_files	\${TWS_HOME}/xtrace ./xcli -snap <snapfile> -p <process> ./xcli -format <snapfile> -d <symbolDB> -xml	<process>.snap_file, <process>.snap_file.xml
<root_dir>/was_info	\${WAS_SERVER_DIR} \${WAS_SERVER_DIR} \${WAS_PROFILE_DIR}/config/cells/ DefaultNode/security.xml "find \${WAS_DIR}/profiles" output "showDataSourceProperties.sh" output "showHostProperties.sh" output "showSecurityProperties.sh" output	\${WAS_SERVER_DIR}_config_listing.txt *.* security.xml websphere_profile_home_list.txt DataSourceProperties.txt (on twcuser = root) HostProperties.txt (on twcuser = root) SecurityProperties.txt (on twcuser = root)
	For Solaris Operating Environment and HP-UX: \${WAS_PROFILE_DIR}/logs/\${WAS_SERVER_DIR}	native_stdout.log
	For all other UNIX: \${WAS_PROFILE_DIR}	javacore*.txt
<root_dir>/was_info/logs	\${WAS_PROFILE_DIR}/logs (WebSphere logs)	*.*
<root_dir>/was_info/ logs/<add. folders>	\${WAS_DIR}/logs (TWS specific logs)	*.*

Table 6. Collected data structure on Windows

Gathered data directory structure	TWS filesystem or command	Files and listings
<root_dir>	General collector output	TWS_%today%_files.txt NODE_<hostname>_TWSUser_<twcuser>_ Base_Date_<yyyymmdd>.README
<root_dir>/db2_oracle_info	"db2support -d <db2db>" output	db2support.zip

Table 6. Collected data structure on Windows (continued)

Gathered data directory structure	TWS filesystem or command	Files and listings
<root_dir>/system_info	"netstat -aboprsv" output "echo %COMPUTERNAME%" output "nslookup %local_cpu%" output %windir%\System32\drivers\etc\hosts "set" output (sc qc tws_maestro %tws_user% output sc qc tws_netman %tws_user% output sc qc tws_tokensrv %tws_user% output) %windir%\System32\drivers\etc\services dir /w "%TMP_DIR%" dir /w "%TWS_HOME%" ntprocinfo.exe -v findstr /I /c:%TWS_HOME%	cpu_netstat_info.txt cpu_node_info.txt cpu_nslookup_info.txt hosts instance_env_info.txt local_services_info.txt services tmp_disk_available.txt tws_home_disk_available.txt tws_process_listing.txt
<root_dir>/tws_info	%TWS_HOME%\Symphony %TWS_HOME%\Sinfonia %TWS_HOME%\StartUp.cmd %TWS_HOME%\Jnext*. * %TWS_HOME%\prodsked %TWS_HOME%\Symnew %TWS_HOME%\Jobtable %TWS_HOME%\schedlog %TWS_HOME%\localopts %TWS_HOME%\mozart\globalopts %TWS_HOME%\bin\dumpsec %TWS_HOME%\jobmanrc.cmd %TWS_HOME%\djobmanrc.cmd dir %TWS_HOME%* dir %TWS_HOME%\bin* dir %TWS_HOME%\mozart* dir %TWS_HOME%\pids* dir %TWS_HOME%\network* dir %TWS_HOME%\audit\database* %TWS_HOME%\audit\database\%today% %TWS_HOME%\audit\database\%yesterday% %TWS_HOME%\audit\plan %TWS_HOME%\audit\plan\%today% %TWS_HOME%\audit\plan\%yesterday% %TWS_HOME%\BmEvents.conf %TWS_HOME%\BmE* Composer output on master %WINDIR%\system32\TWSRegistry.dat %WINDIR%\system32\TWSZOSConnRegistry. datxcopy /S "%TWS_HOME%\version	Symphony Sinfonia StartUp.cmd Jnext*. * prodsked Symnew Jobtable M%today%*, M%tomorrow%*, M%yesterday%* (-date option used) M%yesterday% (-date option not used) localopts globalopts Security_file.txt jobmanrc.txt djobmanrc.txt twshome_files_list.txt tws_binary_list.txt mozart_dir_list.txt pids_dir_list.txt network_dir_list.txt audit_database_dir_list.txt audit_database_%today% audit_database_%yesterday% audit_plan_dir_list.txt audit_plan_%today% audit_plan_%yesterday% BmEvents.conf BmEvents_event_log.txt job_defs, sched_defs, cpu_defs, calendar_defs, parms_defs, resource_defs, prompt_defs, user_defs TWSRegistry.dat TWSZOSConnRegistry.dat *. *
<root_dir>/tws_<version>_install	xcopy /S "%TEMP%\tws\TWS_VMR%" xcopy /S "%TEMP%\tws\TWS_VMR%fixpack"	*. * *. *
<root_dir>/tws_ita_files	dir %TWS_HOME%\ITA %TWS_HOME%\ITA*.ini %TWS_HOME%\ITA*.log %TWS_HOME%\ITA*.out	ita_dir_list.txt *.ini *.log *.out,
<root_dir>/tws_jobmgr_ffdc_files	%TWS_HOME%\stdlist\JM\JOBMANAGER-FFDC*	*. *
<root_dir>/tws_jobmgr_files	dir %TWS_HOME%\stdlist\JM %TWS_HOME%\stdlist\JM*	jobmanager_dir_list.txt *. *
<root_dir>/tws_jobstore_files	dir %TWS_HOME%\jmJobTableDir %TWS_HOME%\jmJobTableDir*	jobstore_dir_list.txt *. *
<root_dir>/tws_logs	%TWS_HOME%\stdlist/<date> %TWS_HOME%\stdlist/logs %TWS_HOME%\stdlist/traces	twsuser, netman and batchup files from date and date-1 twsmerge and netman logs from date and date-1 twsmerge and netman traces from data and date-1

Table 6. Collected data structure on Windows (continued)

Gathered data directory structure	TWS filesystem or command	Files and listings
<root_dir>/tw_ methods	%TWS_HOME%\methods\ echo %CMDEXTVERSION% (PeopleSoft method) psagent.exe -v (PeopleSoft method) r3batch -v (SAP method)	*.*, CMDEXTVERSION.txt psagent_exe_v.txt r3batch_ver.txt
<root_dir>/tw_ msg_ files	"%TWS_HOME%*.msg" "%TWS_HOME%\pobox*.msg" dir "%TWS_HOME%\ftbox*"	*.msg, msg_file_listing.txt *.msg ftbox_dir_list.txt
<root_dir>/tw_ methods	%TWS_HOME%/methods ./r3batch -v ./r3batch -t PL -c <cpu> -l * -j * -- "-debug -trace"	*, methods_dir_list.txt <cpu>_r3batch_ver.txt, r3batch version output <cpu>_r3_batch_info.txt, picklist of scheduled jobs on SAP
<root_dir>/tw_ xtrace_files	%TWS_HOME%/xtrace xccli -snap <snapfile> -p <process> xccli -format <snapfile> -d <symbolDB> -xml	<process>.snap_file, <process>.snap_file.xml
<root_dir>/was_ info	dir "%WAS_SERVER%*" dir "%WAS_HOME%\profiles %WAS_PROFILE%\config\cells\DefaultNode\ security.xml @cmd /C "%WAS_TOOLS%\showDataSource Properties.bat" @cmd /C "%WAS_TOOLS%\showHost Properties.bat" @cmd /C "%WAS_TOOLS%\showSecurity Properties.bat"	%WAS_SERVER%_config_listing.txt websphere_profile_home_list.txt security.xml showDataSourceProperties.txt showHostProperties.txt showSecurityProperties.txt
	%WAS_PROFILE%	javacore*.txt
<root_dir>/was_ info/ %WAS_SERVER%_config_files	%WAS_SERVER%*	*,*
<root_dir>/was_ info/ MAIN_WAS_LOGS	xcopy /S "%WAS_HOME%\logs" (WebSphere logs)	*,*
<root_dir>/was_ info/ %WAS_PROFILE%_logs	xcopy /S "%WAS_PROFILE%\logs" (TWS specific logs)	*,*
<root_dir>/was_ info/ %WAS_PROFILE%.deleted	xcopy /S "%WAS_HOME%\profiles\ %WAS_PROFILE%.deleted"	*,*

First failure data capture (ffdc)

Describes how the data capture tool is used automatically by components of the product to create a *first failure data capture* of the products logs, traces and configuration files.

To assist in troubleshooting, several modules of the product have been enabled to create a *first failure data capture* in the event of failure. This facility uses the data capture tool tws_inst_pull_info (see "Data capture utility" on page 41) to copy logs, traces, configuration files and the database contents (if the database is on DB2) and create a zip which you can send to IBM Software Support.

This tool is run in the following circumstances:

Jobman fails

If batchman detects that jobman has failed, it runs the script, placing the output in <TWA_home>/stdlist/yyyy.mm.dd/collector/JOBMAN

Batchman fails

If mailman detects that batchman has failed, it runs the script, placing the output in <TWA_home>/stdlist/yyyy.mm.dd/collector/BATCHMAN

Mailman fails

If mailman detects that it itself has failed with a terminal error, it runs the script, placing the output in `<TWA_home>/stdlist/yyyy.mm.dd/collector/MAILMAN`. Note that process hard stops, for example, segmentation violations, are not tracked by mailman itself.

Netman child process fails

If netman detects that one of its child processes has failed, it runs the script, placing the output in `<TWA_home>/stdlist/yyyy.mm.dd/collector/NETMAN`

Only one data collection is kept for each day. Each day a new data collection overwrites the previous day's collection.

Within each of the target output directories, the output file is stored in the `/twc_info/TWC_yyyymmdd_hhmmss` directory.

To perform ffdc, the `twc_inst_pull_info` script is run by a script called `collector.sh` (.cmd). You can customize this script (located in `<TWA_home>/TWC/bin`) to apply different parameters to the `twc_inst_pull_info` script for any of the enabled modules (jobman, mailman, batchman and netman)

Creating a core dump of the application server

If the WebSphere Application Server hangs, and you decide to contact IBM Software Support for assistance, it would help the diagnosis of the problem if you could provide one or more core dumps taken during the hang. Use the following procedure to create a core dump:

1. Log on as a WebSphere Application Server administrator
2. Change to the directory: `WAS_profile_path/bin` and run the script **wsadmin.sh/bat** to open the administration shell. `WAS_profile_path` corresponds to the WebSphere Application Server profile path you specified at installation time. The default path is `TWA_home/WAS/TWSpfile`.

3. Set the `jvm` variable as follows:

```
set jvm [$AdminControl completeObjectName type=JVM,process=<server_name>,*]
```

where `<server_name>` is determined by looking in the following directory:
`WAS_profile_path/config/cells/ TWSNodeCell/nodes/TWSNode/servers`.

For each instance of Tivoli Workload Automation on the computer you will see a directory, the name of which is the `<server_name>`. If there is more than one directory you must determine which instance you want to dump.

4. Run the core dump as follows:

```
$AdminControl invoke $jvm dumpThreads
```

This creates a core dump in the `WAS_profile_path/bin` directory with the following name:

Windows and Linux

`javacore.<yyyymmdd>.<hhmmss>.<pid>.txt`, where `yyyy` = year, `mm` = month, `dd` = day, `ss` = second, and `pid` = process ID.

UNIX `javacore<pid>.<time>.txt` where `pid` = process ID and `<time>` = the number of seconds since 1/1/1970.

5. Repeat step 4. The more dumps you can take, the more information is available to the support team.

6. Send the dumps, the application server log files and a detailed description of what you were doing, to IBM Software Support.

Chapter 4. In-Flight Trace facility for engine

Describes the tracing facility for troubleshooting the Tivoli Workload Scheduler engine. This facility is called In-Flight Trace.

This document describes the Tivoli Workload Scheduler server tracing facility that replaced Autotrace from version 8.6. The facility is designed to be used by IBM Software Support, but is fully described here so that you understand how to use it if requested to do so by IBM Software Support.

The Tivoli Workload Scheduler server tracing facility (hereafter called In-Flight Trace) is a facility used by IBM Software Support to help solve problems in Tivoli Workload Scheduler. At maximum capacity it can trace the entry into and exit from every Tivoli Workload Scheduler function, plus many other events, and includes all log and trace messages currently issued by the CCLog facility.

In-Flight Trace has been conceived as a multi-product tool, although this description concentrates on its use for Tivoli Workload Scheduler.

It works as follows:

Existing trace calls

In-Flight Trace uses the logging and tracing facilities still used by the CCLog logging and tracing mechanism, and which were used by the Autotrace facility in releases before 8.6.

Function entry and exit

In addition, the Tivoli Workload Scheduler engine product build now inserts trace calls in the code to record the entry to and exit from every function and assigns a sequential numeric function ID to each function. The trace calls use these IDs to identify the functions.

Building the xdb.dat symbols database

During the same process, the build creates the **xdb.data** symbols database associating the name of each function with the function ID. In this way, the trace writes the minimum information possible to the trace record (the function ID), which can then be expanded to give the function name later for viewing.

The build also stores in the database the source file and line number of each function.

Further, it stores the name of the component which "owns" the function. One program contains many components, each of which contains many functions.

The symbols database is the key to managing the activation/deactivation and filtering of the traces. The information it contains is encrypted.

Tracing in shared memory

The traces are written to shared memory. This is divided into segments, and the traces chosen to be written to each segment are written in an endless loop. At maximum capacity (tracing all events on all functions) the traces might loop every few seconds, while at minimum capacity (tracing just one little-used function), the trace might not loop for months.

Segments

You can choose to use any number of segments (each is identified by a unique number) and for each segment can determine how much shared memory the segment is to use. More and bigger segments consume more memory, with all the normal consequences that entails.

Programs

Any number of Tivoli Workload Scheduler programs can be configured to be saved to the same segment. You decide which programs are to be traced to which segments, and whether those segments are to be enabled for tracing, by modifying the basic configuration. Any of the Tivoli Workload Scheduler programs and utilities can be configured for tracing.

Basic configuration

The basic configuration determines which segments are enabled for tracing, and makes an initial determination of whether the tracing for a specific program is activated. It is achieved by editing a configuration file with a text editor. The Tivoli Workload Scheduler engine (the product) must be restarted to make the changes take effect. The configuration is divided into the following sections:

Global

This section not only includes general information like the product code and the segment size, but also acts as a "catch-all", where traces from programs not specifically configured are configured.

<program>

If a program is not to be traced under the "global" section, a specific program section must be configured, defining which segment the program is to be traced in, and other basic information. The information in a program section overrides that in the global section, but just for that program.

Activating and deactivating traces

For segments which are enabled, traces for specific programs can be activated and deactivated on-the-fly, from the command line, as these flags are held in memory.

Trace levels

Events in the code have been assigned trace levels. The lower the level, the more drastic the event. The levels range from reporting only unrecoverable errors, through recoverable errors, warnings, and informational messages and three debug levels to the maximum reporting level, where even function entry and exit events are recorded.

Trace levels can also be changed on-the-fly, from the command line, without restarting the engine.

Snapshots

In-Flight Trace lets you take a snapshot of the current contents of the traces for a program or segment and save it to a file. You can optionally clear the memory in the segment after taking the snapshot. The snapshot file is in the internal format, containing function IDs, etc., and is not easily readable. It must be formatted to make it readable.

Formatting the snapshot

A command-line option lets you format a snapshot file for the standard output. The output can be in CSV or XML format, and information about the source data (file name and line number) is automatically included. Or you can select the standard trace format (one line per trace record) and

choose whether to include the source information. And finally you can choose whether to include the header information (ideal for a printed output) or not (ideal for the creation of a file you are going to analyze programmatically).

Filtering

The tooling-up of the code is a fully automatic process and you might find that your traces include frequently used components or functions that are not causing any problems. You would like to exclude them from the trace and you do this by using the command line to create a filter file, in which you can specify to include all and then exclude any combinations of specific components, functions, and source files. Alternatively, you can exclude all and then include any combinations of specific components, functions, and source files. Functions can also be included or excluded by specifying a range of function IDs.

Once created, a filter file is declared either in the global section of the configuration file or one of the program sections. You can have more than one filter file which you use with different programs, however, note that the filter is applied at segment level. This means that if you have two programs writing to the same segment, the filter is applied to both even if it is only specified for one of them.

Existing filter files can be modified from the command line.

Products

In-Flight Trace is conceived as a multi-product facility. Each product has its own separate configuration file. Multiple instances of the facility can be run on the same system, completely independently of each other. However, you can also control one product from the tracing facility of another, by identifying the product to which to apply the commands. For example, if you had two versions of Tivoli Workload Scheduler running on the same system, you could control the In-Flight Trace facility for both of them from one place, inserting the appropriate product code when required by the command syntax.

In-Flight Trace configuration file

Describes the In-Flight Trace configuration file, `xtrace.ini`.

The In-Flight Trace configuration file is used to initialize the shared memory at product startup. The information in shared memory determines which traces are saved at which level. All function trace calls are parsed by the trace facility to determine if they should be saved.

The In-Flight Trace configuration file is found in the following path:

`<TWA_home>/TWS/xtrace/xtrace.ini`

An example of the file is as follows:

```
[ _GLOBAL_ ]
Product      = <PRODUCT>
Enabled      = y
Active       = y
SegNum       = 1
FilterFile   = $(install_dir)/bin/xfull.xtrace
SegSize      = 10240
Level        = 80
SegPerUser   = n
```

```
[netman]
Enabled      = y
Active       = y
SegNum       = 2
Level        = 80
```

Changing the configuration

Describes how to modify the configuration file.

To permanently change the configuration in shared memory that controls the tracing, edit the file, save it, and restart the product. On UNIX platforms you must also clean up the memory by running the tracing command with the **-clean** parameter between stopping and restarting the product. Thus, the procedure to change the configuration file is as follows:

UNIX

1. Modify the configuration file
2. Save the configuration file
3. Stop the product
4. Run **xcli -clean**
5. Restart the product

Windows

1. Modify the configuration file
2. Save the configuration file
3. Stop the product
4. Restart the product

You can change much of the configuration in shared memory that controls the tracing by using the **xcli** command (see “xcli command syntax” on page 59). However, any changes made in this way are not updated in the configuration file, so at the next initialization, unless you have specifically edited the file, the parameters used are those that were in the file last time you restarted the product.

Configuration file syntax

Describes the syntax of the configuration file.

The file is divided into sections. Each section begins with a section header in one of the two following formats (the square brackets are required - they are not command syntax indicators):

```
[ _GLOBAL_ ]
[<program>]
```

[_GLOBAL_]

There must be only one [_GLOBAL_] section containing general information about the product and the tracing configuration for all programs that do not have a specific section.

[<program>]

You can define a separate section for each of the Tivoli Workload Scheduler programs ([<program>]). The following are the programs that are most likely to require tracing:

```
APPSRVMAN
BATCHMAN
```


JOBMAN
JOBMON
MAILMAN
NETMAN
WRITER
JAVA (the connector)

However, you can trace any executable program such as COMPOSER, CONMAN and all the utilities - in fact any program stored in the Tivoli Workload Scheduler /bin directory.

You cannot have more than one instance of a section for the same program.

If a program has no specific section, its trace configuration uses the defaults in the [_GLOBAL_] section. Details defined in the program sections in almost all cases override the corresponding values in the [_GLOBAL_] section (the exception is Product).

The program name is not case-sensitive. For example, you can write Netman, NetMan, netman or NETMAN.

Note: On UNIX operating systems, JOBMAN and jobman are two separate programs performing different functions. This means that on UNIX operating systems, because of the case-insensitivity, if you set a trace configuration section for either JOBMAN or jobman, both programs will be traced using that section and therefore to the same segment. This is a limitation that cannot be avoided at present.

Available keys (each key can be defined only once in each section):

Product

[_GLOBAL_] only. Product identification string. Required.

Enabled

Specifies if the segment is enabled. If you change the enablement of a segment by changing this value and saving the configuration file, you must restart the product to make the change effective. If the segment in the [_GLOBAL_] section is not enabled, the entire tracing facility is disabled. Enter "y" or "n".

Active Specifies whether tracing for the specific program is active. If the [_GLOBAL_] section is not activated, the tracing for all programs without a specific section is not activated. This value can be changed without restarting the product by using the tracing command. Enter "y" or "n".

SegNum

Determines the segment number to use for tracing for a specific section. More than one program can be defined for the same segment in different sections. The SegNum specified in the [_GLOBAL_] section is used by any program that does not have a specific section defined. If you change the segment number of a program by changing this value and saving the configuration file, you must restart the product to make the change effective. Enter any numeric value.

FilterFile

Specifies the file that contains the criteria for filtering components, functions, or source files. The file is applied at segment level, so you cannot specify different filter files for different programs that use the same

segment. This value can be changed without restarting the product by using the tracing command. Enter the fully qualified file path.

The default filter file supplied with the product does not trace the top 5% most-used routines (by being most-used they are less likely to exhibit problems).

SegSize

Specifies the segment size (Kb). If this value is supplied more than once in different sections for the same segment, the trace facility uses the highest of the supplied values. If you change the size of a segment by changing this value and saving the configuration file, you must restart the product. Enter a numeric value.

The full shared memory usage is the sum of all enabled segments, plus several Kbytes for the control data.

On UNIX, ensure that you do not exceed the configurable kernel parameter which determines the maximum size of shared memory.

Level Specifies the maximum level to be traced for the program. Enter one of the following specific numeric values:

Level	Description
10	Unrecoverable
20	Error
30	Warning
40	Informational
50	Debug minimum
60	Debug medium
70	Debug maximum
80	Function entry and exit

If this value is supplied more than once in different sections for different programs that trace in the same segment, the trace facility uses the appropriate values for each program. Thus, the segment might contain traces for one program at level 10 and for another at level 80.

This value can be changed without restarting the product by using the tracing command.

SegPerUser

Specifies if the segment provides access to the owner of the segment only (y) or all users (n). Enter "y" or "n"

In-Flight Trace command: xcli

Describes the In-Flight Trace **xcli** command.

This topic describes the command used to control all aspects of the runtime behavior of In-Flight Trace.

The command modifies the information in shared memory. Shared memory is initialized from the information in the configuration file, but any changes to shared memory that are made using the options of this command are *not* saved in the configuration file.

Selecting programs, segments, and products

Describes how to select programs, segments, and products in the `xcli` command.

In many of the parameters of the `xcli` command, you are required to select a program or a segment, and optionally a product. To avoid repeating the same information, details of how to do this are supplied here:

Program

Select a program for a specific action by identifying the global section ([`_GLOBAL_`]) or any of the configuration file sections containing Tivoli Workload Scheduler programs ([`<program>`]).

Segment

Select one of the segment numbers that were defined in the configuration file when the shared memory was initialized. If you need to use extra segments or redistribute the programs within the segments, you must edit and save the configuration file and then stop and restart the Tivoli Workload Scheduler engine.

Product

The tracing facility is multi-product. However, if you run the `xcli` command from the same directory as a configuration file, you automatically run it on the product defined in that configuration file, without having to define the product in the command.

But if you are using In-Flight Trace to trace more than one product, and you want to use the command supplied with product A to modify the tracing of product B, you must supply the product code for product B as a parameter to the command, by adding the `-P <product>` parameter to the command string. This parameter is only applicable to the `-snap`, `-query`, `-active`, `-level`, and `-filter` subcommands.

xcli command syntax

Gives the full syntax of the `xcli` command.

Controls all of the runtime aspects of the In-Flight Trace facility. It modifies the information in shared memory. Shared memory is initialized from the information in the configuration file, but any changes to shared memory made using the options of this command are *not* saved in the configuration file.

You must be the `TWS_user` to run the command.

Syntax

`xcli`

```
-snap <snap_file>
{ -p <program> | -s <segment> }
[ -descr <description> ]
[ -clean ]
[ -P <product> ]

-format <snap_file>
-d <symbols_database>
[ -full ]
[ -noHeader ]
[ -standard [ -source ] | -xml | -csv ]
```

```

-query [ -p <program> | -s <segment> ] [-P <product> ]

-active { y | n }
    { -p <program> | -s <segment> | -all }
    [ -P <product> ]

-level <level>
    { -p <program> | -s <segment> | -all }
    [ -P <product> ]

-filter <filter_file>
    { -p <program> | -s <segment> }
    [ -P <product> ]

-createFilter <filter_file> -d<symbols_database>
    [ -add_all |
      -add_comp <component> | -remove_comp <component> |
      -add_func <function_name> | -remove_func <function_name> |
      -add_func_id <function_ID> | -remove_func_id <function_ID> |
      -add_func_id_range <from> <to> | -remove_func_id_range <from> <to> |
      -add_filter <filter_file> | -remove_filter <filter_file> ] ...

-modifyFilter<filter_file> -d<symbols_database>
    [ -add_all | -remove_all
      -add_comp <component> | -remove_comp <component> |
      -add_func <function_name> | -remove_func <function_name> |
      -add_func_id <function_ID> | -remove_func_id <function_ID> |
      -add_func_id_range <from> <to> | -remove_func_id_range <from> <to> |
      -add_filter <filter_file> | -remove_filter <filter_file> ] ...

-clean

-config [<config_file> ]

```

Arguments

-snap <snap_file>

Saves a snapshot of part of the shared memory to the indicated file. The following are the snapshot parameters:

{ **-p** <program> | **-s** <segment> }

Define if the snapshot is for either a program or a segment. If it is made for a program which shares a segment with other programs, the whole segment is snapped, but the header information shows which program it was snapped for. See also “Selecting programs, segments, and products” on page 59.

[**-descr** <description>]

Supply a description for the snapshot. Surround it with double quotation marks if it contains spaces.

[**-clean**]

Optionally clear the entire segment memory after taking the snapshot. If any process is still using the memory, the clean operation cannot be performed and a warning message is given.

Note: If your snapshot is of a program, this option clears the memory for all traces in the segment for which the program is configured, including those of any other programs that have been configured to write to it.

[**-P** *<product>*]

See “Selecting programs, segments, and products” on page 59.

The snap file header information is as follows:

```
"Snap information:
" Product:      <product>
" Description:  <description>
" Snap platform: <platform>
" Snap time (GMT): <time>
" Snap program: <program>
" Snap segment: <segment>
"   Segment size: <size>(Kb)
"   Segment use: <percent_used>
```

-format *<snap_file>*

Formats the supplied snapshot file for the standard output. The formatting options are:

-d *<symbols_database>*

Supply the name of the symbols database to use for the formatting. The database must be either the same version as the instance of Tivoli Workload Scheduler from which the snap was captured (ideally), or a later version. The default symbols database is **xdb.dat**.

[**-full**]

If the snap was taken of a single program in a multi-program segment, use this option to send the full set of traces (all programs) to the standard output, rather than that of the single program as determined by the header information in the snap file.

[**-noHeader**]

Use this to suppress the output of the header information. The standard output then just consists of trace messages, which is more acceptable as input to an analysis program.

[**-standard** [**-source**] | **-xml** | **-csv**]

Define the formatting of the traces. If you have selected **-standard**, use the optional parameter **-source** to add information about the source file and line number. This source information is automatically included in the **-xml** and **-csv** options. If you supply none of these, the format defaults to **-standard**.

-query Outputs the enablement or activation state of a program or segment. Without parameters, this option displays information about the entire configuration to the standard output. The parameters are:

[**-p** *<program>* | **-s** *<segment>*]

Optionally define whether the query is for a specific program or a specific segment. See also “Selecting programs, segments, and products” on page 59.

[**-P** *<product>*]

See “Selecting programs, segments, and products” on page 59.

-active { *y* | *n* }

Activates (*y*) or deactivates (*n*) a program or segment in memory, or all programs and segments. The parameters are as follows:

{ **-p** <program> | **-s** <segment> | **-all** }

Activate either a specific program or a specific segment, or all programs and segments. See also “Selecting programs, segments, and products” on page 59.

[**-P** <product>]

See “Selecting programs, segments, and products” on page 59.

-level <level>

Sets the tracing level for a program or segment in memory. Supply one of the following level codes:

Level	Description
10	Unrecoverable
20	Error
30	Warning
40	Informational
50	Debug minimum
60	Debug medium
70	Debug maximum
80	Function entry and exit

For example, to trace only unrecoverable failures and errors, supply "20".

The parameters are as follows:

{ **-p**<program> | **-s**<segment> | **-all** }

Set the level for either a specific program or a specific segment, or all programs and segments. See also “Selecting programs, segments, and products” on page 59.

[**-P**<product>]

See “Selecting programs, segments, and products” on page 59.

-filter <filter_file>

Applies a new filter file for a program or segment in shared memory. The parameters are as follows:

{ **-p** <program> | **-s** <segment> }

Determine the filter file to be used for either a program or a segment. See also “Selecting programs, segments, and products” on page 59.

[**-P** <product>]

See “Selecting programs, segments, and products” on page 59.

default

The filter file is created using the **-createFilter** option. In this option (and the associated **-modifyFilter** option) you specify any components and functions you want to include or exclude from the tracing (see below for more details). This information is written in the filter file as a list of all functions in the symbols database (by ID) with a bit set to indicate whether they are to be included or excluded. The default symbols database is **xdb.dat**.

Any filter files defined in the configuration file are loaded into shared memory at initialization. If you use this option, the shared memory area is overwritten with the new contents. If the new filter file has been created

using a different symbols database than the original file, a warning is given, because it is advisable to use the same symbols database when creating the new filter file.

The default filter file supplied with the product is set to not trace the 5% most-used routines, on the basis that the most-used routines are less likely to create problems because they are well tried and tested.

-createFilter <filter_file>

Creates the filter file named in the parameter. The file must not already exist. There is no facility to view a filter file, so use meaningful names and maintain your own documentation of the contents of each filter file.

To populate the file supply one or more of the following parameters. If you *add* an item, its traces will be saved; if you *remove* an item, its traces will *not* be saved. By default, all components and functions are removed.

-d <symbols_database>

Identify the symbols database to use to verify the component names, and the function names and IDs.

-add_all

Add all components and functions to the filter file. Use this with one of the **-remove** options to create an exclusive "all except ..." filter.

-add_comp <component> | **-remove_comp** <component>

Add a component to the file or remove one that has already been added. For example, you could add all components using **-add_all** and then remove just one, which would be easier than adding all of the required components individually. Discover component names by viewing a formatted snapshot.

-add_func <function_name> | **-remove_func** <function_name>

Add a function to the file or remove one that has already been added. For example, you could add a component using **-add_comp** and then remove one of its functions, which would be easier than adding all of the required functions individually. Discover function names by viewing a formatted snapshot.

-add_func_id <function_ID> | **-remove_func_id** <function_ID>

Adds a function to the file by ID, or removes one that has already been added. For example, you could add a component using **-add_comp** and then remove one of its functions, which would be easier than adding all of the required functions individually. A function ID is a sequential number allocated to a function when the product was built, and stored in the symbols database. Discover function IDs by viewing a formatted snapshot.

-add_func_id_range <from> <to> | **-remove_func_id_range** <from> <to> |

Adds a range of functions to the file by ID or removes a range of functions that have been already added. Discover function IDs by viewing a formatted snapshot.

-add_filter <filter_file> | **-remove_filter** <filter_file>

Adds or removes the contents of an existing (different) filter file, as follows:

Adding a filter file

If you add a filter file, the items in that filter file which are set to be filtered (traced) are added to whatever other filter criteria you might have set.

Removing a filter file

If you remove a filter file, the items in the filter file which are set to be filtered (traced) are removed from whatever other filter criteria you might have set.

For example, you might create a filter file that configures the tracing of the communications functions. You could then add this set of functions to your filter set in one command, or remove them, depending on whether you think the communications are part of the problem you are trying to solve.

The add and remove actions are processed in the order you submit them. Thus if you add a function ID, and then remove a range that includes that ID it is removed from the criteria. But if you remove the range, and then add the ID, it is added to the criteria.

-modifyFilter <filter_file>

Modifies the existing filter file named in the parameter.

This subcommand takes all of the parameters used in the **-addFilter** subcommand in the same way, with the addition of the following:

-remove_all

Removes all components and functions from the filter file. Use this with one of the **-add** options to create an inclusive "all of the following" filter.

-clean On UNIX operating systems only, use this to delete the shared memory segments after you have modified and saved the configuration file, and stopped the product. If a segment is in use, it is marked for deletion and will be automatically deleted when no longer in use.

-config [<config_file>]

This initializes the memory. It is run automatically when the Tivoli Workload Scheduler engine is restarted, using the default configuration file `./xtrace.ini`. In normal circumstances, you never need to run this manually. If you believe that the shared memory is corrupted, it is better to restart the product, which automatically re-initializes the memory.

Examples

The following examples are a scenario for using the trace to troubleshoot an instance of Tivoli Workload Scheduler which is hanging for 5 minutes when you run a particular utility command, without giving any log messages to indicate why.

The presupposition is that you have the following configuration file:

```
[ _GLOBAL_ ]
Product      = TWS_8.6.0
Enabled      = y
Active       = n
SegNum       = 1
FilterFile   = $(install_dir)/bin/xfull.xtrace
SegSize      = 10240
Level        = 80
SegPerUser   = n

[netman]
Enabled      = y
Active       = n
```



```
SegNum      = 2
Level       = 80
```

```
[batchman]
Enabled     = y
Active      = n
SegNum      = 3
Level       = 80
```

1. Start the tracing

Tracing is enabled but inactive for three segments. You think the problem is not network related, so **netman** is not involved. To activate the other two segments run the following commands:

```
xcli -active y -s 1
xcli -active y -s 3
```

2. Adjust the levels for minimum debug

You want to trace as much activity as possible so that you understand what is happening. So you adjust the tracing levels to minimum debug:

```
xcli -level 50 -s 1
xcli -level 50 -s 3
```

3. Take a snapshot when the product hangs

Restart Tivoli Workload Scheduler and run the utility again. When the product hangs immediately take a snapshot of each segment. You include the option to clean the memory after the snapshot:

```
xcli -snap main_snap -s 1
      -descr "Snap of segment 1 when TWS hangs after using utility" -clean
xcli -snap batchman_snap -s 3
      -descr "Snap of batchman when TWS hangs after using utility" -clean
```

4. Format the trace to view it

Run the following command for a standard format for each file, and save it to a text file:

```
xcli -format main_snap -d xdb.dat > main_snap.txt
xcli -format batchman_snap -d xdb.dat > batchman_snap.txt
```

5. The problem seems to be with batchman, but you need more detail

After examining the two snap files it seems as though the problem is occurring in batchman, but you need more detail:

```
xcli -level 80 -s 3
```

6. Take another snapshot of batchman when the product hangs

Restart Tivoli Workload Scheduler and run the utility again. When the product hangs immediately take another snapshot of batchman's segment:

```
xcli -snap batchman2_snap -s 3 -descr "Second snap of batchman (level 80)"
```

7. Format the trace again to view it

Run the following command to save the snap file in XML format to a file:

```
xcli -format batchman2_snap -d TWS86SymDB -xml > batchman2_snap.xml
```

You now have a well-formatted XML file of the traces to examine in details and determine where the problem is occurring.

xcli messages

Lists all the messages that might be issued by the xcli command.

This section details the messages that might be produced by xcli and explains what they mean.

Incorrect syntax in configuration file.

In-Flight Trace has found syntax that it cannot parse in the configuration file. Check the syntax carefully with the information in this manual. Correct the error and rerun the command.

Cannot create the semaphore '%d', error %ld.

The error message is from the operating system. There might be a memory usage problem which requires an operating system reboot.

Cannot lock the semaphore, error %ld.

The error message is from the operating system. There might be a memory usage problem which requires an operating system reboot.

Cannot create the shared memory, error %ld.

The error message is from the operating system. There might be a memory usage problem which requires an operating system reboot.

Cannot map the shared memory, error %ld.

The error message is from the operating system. There might be a memory usage problem which requires an operating system reboot.

Incorrect value for key %s in section '%s'.

The syntax of the configuration file is correct but the indicated key in the indicated section has an incorrect value.

The tracing facility is not active.

If the "Enable" key in the [_GLOBAL_] section is set to "n", the tracing facility is disabled. To enable it, edit the configuration file, set the "Enable" key in the [_GLOBAL_] section to "y", save the file and restart Tivoli Workload Scheduler.

Unable to open file '%s', error %d.

The error message is from the operating system. Check the error code. The file might be open in another process, or the user running the command might not have rights to open the file. Correct the problem and try the command again.

Not enough free memory to allocate %d bytes.

The message indicates the memory required by your configuration. Either reduce the amount of memory used by the configuration by editing the configuration file, changing the values, saving the file and restarting Tivoli Workload Scheduler, or free some memory by closing other applications. You might also be able to enlarge the memory paging file. Use the **-config** option to reinitialize the memory.

Unable to write to file '%s', error %d.

The error message is from the operating system. Check the error code. The file might have been deleted by another process, or the user running the command might not have rights to write to the file. Correct the problem and try the command again.

Unable to read from file '%s', error %d.

The error message is from the operating system. Check the error code. The file might have been deleted by another process, or the user running the command might not have rights to read from the file. Correct the problem and try the command again.

The selected file does not contain a valid snapshot.

You have identified a snapshot file to format, but either it is not a snapshot file or the snapshot file was not written correctly. Check the name you

supplied. If it was not correct, reissue the command with the correct file name. If the file name is correct, rerun the snap to regenerate the file.

Memory not correctly initialized.

The shared memory has not been correctly created. Check that there is sufficient free memory to create the shared memory you have defined in the configuration file.

The tracing facility is not active for program %s.

You have requested to change tracing information for the indicated program which is not active. Activate the program first, using the **-active** option.

The tracing facility is not active for segment %s.

You have requested to change tracing information for the indicated segment which is not active. Activate the segment first, using the **-active** option.

Operation successful.

No-brainer! Whatever you were doing has worked!

Unable to remove the semaphore %x, error %d.

The error message is from the operating system. There might be a memory usage problem which requires an operating system reboot.

Unable to remove the shared memory %x, error %d.

The error message is from the operating system. There might be a memory usage problem which requires an operating system reboot.

The tracing facility is not active for product %s.

You have identified a product for which the "Enable" key in the [_GLOBAL_] section is set to "n", and so the tracing facility is disabled. To enable it, edit the appropriate configuration file, setting the "Enable" key in the [_GLOBAL_] section to "y", save the file and restart the product.

The maximum number of products (%d) has already been reached.

In-Flight Trace can only trace a limited number of products at one time, regardless of the amount of memory available. You have reached that limit!

The sections '%s' and '%s' have the same segment number but different %s.

Some of the keys in a section are "segment-based", in that if more than one section traces to the same segment, they must have the same values. For example, the filter file for programs that trace to the same segment must be the same. Either change the programs to trace to different segments or supply the same filter file for all programs that trace to the same segment.

Segment %d is too small for the filter file '%s'.

The space you allocate to a segment is used to store the filter file for more than 12,000 functions, in addition to the traces. In this case, you have not created sufficient space to store the filter file. You cannot change the size of the filter file, because it uses one entry for every function in the product, regardless of whether that function is or is not filtered for tracing. So you must increase the segment size by editing the configuration file, saving it and restarting Tivoli Workload Scheduler.

Cannot open the symbols database '%s'.

Either the symbols database does not exist with the name you supplied, the user running the command does not have the rights to open the file, or the file is corrupted. Check that the name is correct and ensure that you are the *TWS_user*.

Too many input parameters.

The syntax of the command you supplied is not correct. Check the syntax with what is documented in this publication and try the command again.

An error occurred while opening the symbols database '%s'.

The database file might be corrupted.

Warning: the function ID %d is not in the symbols database.

You have tried to add or remove a function which is not in the symbols database. Check the source from which you obtained the function name or ID. Check that you are using the correct symbols database. The default symbols database is **xdb.dat**. Correct the error and try the command again.

Duplicated section '%s' in the configuration file.

Each program section can only be present once in the configuration file. Perhaps you copied a section intending to change the name but did not. Edit the configuration file, save it and restart Tivoli Workload Scheduler.

Warning: There is a mismatch between the size of the new filter file and the previous one (new size = %d, previous size = %d).

You have used the **-filter** option to supply a new filter file, but the new filter file was generated using a symbols database different from that used when the filter file currently in use was created. Put a different way, you seem to have used different symbols databases to create two different filter files, and the two different databases have different numbers of functions. In-Flight Trace can continue tracing but the filtering might not be applied correctly. You are advised always to use only the symbols database generated when the version of the product you are tracing was built.

Cannot clean up the shared memory because some process is currently using it.

You have used the **-clean** option to clean the shared memory, but one or more processes is still using the shared memory, so it cannot be cleaned. Use your system resources to determine which process is using the shared memory, stop it, and retry the **-clean** option.

Chapter 5. Auditing facilities

Describes the audit facilities to track changes in the database and the plan, as well as those that track changes to objects involved in dynamic workload scheduling.

Audit trails are useful to check enforcement and effectiveness of IT controls, for accountability, and vulnerability and risk analysis. IT organizations can also use auditing of security-related critical activities to aid in investigations of security incidents. When a security incident occurs, audit trails enable analysis of the history of activities (who did what, when, where, and how) that occurred prior to the security incident, so appropriate corrective actions can be taken. For these reasons, audit trails might need to be archived and accessible for years.

The auditing logs are created in XML format and can be viewed with a standard text editor or parsed using third-party utilities.

You can also view the auditing logs using the Log and Trace Analyzer (LTA), a component of the IBM Autonomic Computing Toolkit. In general, the Log and Trace Analyzer is used for importing and correlating different logs generated by different products. The Log and Trace Analyzer can be very useful in correlating auditing logs with other logs from different sources, such as databases (DB2, Oracle), WebSphere Application Server, and the operating system. See “Engine Log Analyzer” on page 18 for details.

Two separate audit trail facilities are provided:

- Database and plan change tracking - see “Database and plan audit”
- Tracking of changes to scheduling objects to support dynamic workload scheduling - see “Dynamic workload scheduling audit” on page 75

Database and plan audit

An auditing option is available to track changes to the database and the plan. It is disabled by default. It is described in these sections:

- “How audit works”
- “Enabling the audit feature” on page 70
- “Audit log header format” on page 70
- “Audit log body format” on page 71
- “Sample audit log entries” on page 74

How audit works

The storage of audit records varies depending on whether you maintain trails for the database or the plan. You have the following options:

database auditing

You can track changes to the database in a file, in the database itself, or in both. All user modifications are logged, including the current definition of each modified database object. If an object is opened and saved, the action is logged even if no modification has been done.

plan auditing

You can track changes to the plan in a file. All user modifications to the plan are logged. Actions are logged whether they are successful or not.

Each audit log provides audit information for one day, from 00:00:00 UTC to 23:59:59 UTC regardless of the timezone of the local workstation, but the log file is only created when an action is performed or the WebSphere Application Server is started.

The files are called `yyyymmdd`, and are created in the following directories:

```
<TWA_home>/TWS/audit/plan  
<TWA_home>/TWS/audit/database
```

Audit entries are logged to a flat text file on individual workstations in the Tivoli Workload Scheduler network. This minimizes the risk of audit failure due to network issues. The log formats are the same for both plan and database in a general sense. The logs consist of a header portion which is the same for all records, an action ID, and a section of data which varies according to the action type. All data is kept in clear text and formatted to be readable and editable from a text editor such as **vi** or **notepad**.

Note: For **modify** commands, two entries are made in the log for resources, calendars, parameters and prompts. The **modify** command is displayed in the log as a combination of the **delete** and **add** commands.

Enabling the audit feature

The auditing option is enabled by setting the following two entries in the global options, using **optman**:

```
enPlanAudit = 0|1  
enDbAudit = 0|1
```

A value of *1* (one) enables auditing and a value of *0* (zero) disables auditing. Auditing is disabled by default on installation of the product.

To initiate database auditing, you must shut down Tivoli Workload Scheduler completely. When you restart Tivoli Workload Scheduler, the database audit log is initiated. Plan auditing takes effect when `JnextPlan` is run.

Audit log header format

Each log file starts with a header record that contains information about when the log was created and whether it is a plan or database log.

The header record fields are separated by vertical bars (`|`), as follows:

```
HEADER|<GMT_date>|<GMT_time>|<local_date>|<local_time>|<object_type>| >  
<workstation>|<user_ID>|<version>| <level>
```

Log Type

HEADER

GMT Date

The GMT date when the log file was created.

GMT Time

The GMT time when the log file was created.

Local Date

The local date when the log file was created. The local date is defined by the time zone option of the workstation.

Local Time

The local time when the log file was created. The local time is defined by the time zone option of the workstation.

Object Type

DATABASE for a database log file and PLAN for a plan log file.

Workstation Name

The Tivoli Workload Scheduler workstation name for which this file was created. Each workstation in the Tivoli Workload Scheduler network creates its own log.

User ID

The Tivoli Workload Scheduler user ID that created the log file.

Version

The version of the file.

Level The logging level.

Audit log body format

The audit log formats are basically the same for the plan and the database. The log consists of a header portion, an action ID, and data sections that vary with the action type. The data is in clear text format and each data item is separated by a vertical bar (|).

The log file entries are in the following format:

```
<log_type>|<GMT_date>|<GMT_time>|<local_date>|<local_time>|<object_type>| >
<action_type>|<workstation>|<user_ID>|<object_name>|<action_data_fields>
```

The log files contain the following information:

log_type

Displays an eight character value indicating the source of the log record. The following log types are supported:

CONMAN

conman command text

DATABASE

Database action

HEADER

The log file header

MAKESEC

makesec run

PARMS

Parameter command text

PLAN Plan action

RELEASE

release command text

STAGEMAN

stageman run

GMT_date

Displays the GMT date the action was performed. The format is *yyyymmdd* where *yyyy* is the year, *mm* is the month, and *dd* is the day.

GMT_time

Displays the GMT time the action was performed. The format is *hhmmss* where *hh* is the hour, *mm* is the minutes, and *ss* is the seconds.

local_date

Displays the local date the action was performed. The local date is defined by the time zone option of the workstation. The format is *yyyymmdd* where *yyyy* is the year, *mm* is the month, and *dd* is the day.

local_time

Displays the local time the action was performed. The local time is defined by the time zone option of the workstation. The format is *hhmmss* where *hh* is the hour, *mm* is the minutes, and *ss* is the seconds.

object_type

Displays the type of the object that was affected by an action, from the following:

DATABASE

Database definition (for header only)

DBCAL

Database calendar definition

DBDOMAIN

Database domain definition

DBJBSTRM

Database Job Scheduler definition

DBJOB

Database job definition

DBPARM

Database parameter definition

DBPROMPT

Database prompt definition

DBRES

Database resource definition

DBSEC

Database security

DBUSER

Database user definition

DBVARTAB

Database variable table definition

DBWKCLS

Database workstation class definition

DBWKSTN

Database workstation definition

PLAN Plan (for header only)

PLDOMAIN
Plan domain

PLFILE
Plan file

PLJBSTRM
Plan Job Scheduler

PLJOB
Plan job

PLPROMPT
Plan prompt

PLRES
Plan resource

PLWKSTN
Plan workstation

action_type

Displays what action was performed on the object. The appropriate values for this field are dependent on which action is being performed.

For the plan, the <action_type> can be ADD, DELETE, MODIFY, or INSTALL.

For the database, the ADD, DELETE and MODIFY actions are recorded for workstation, workstation classes, domains, users, jobs, job streams, calendars, prompts, resources and parameters in the database.

The <action_type> field also records the installation of a new Security file. When **makesec** is run, Tivoli Workload Scheduler records it as an INSTALL action for a Security definition object.

LIST and DISPLAY actions for objects are not logged.

For parameters, the command line with its arguments is logged.

workstation

Displays the Tivoli Workload Scheduler workstation from which the user is performing the action.

user_ID

Displays the logon user who performed the particular action. On Windows operating systems, if the user who installed WebSphere Application Server was a domain user, for Log Types **stageman** and **conman** this field contains the fully qualified user ID *domain\user*.

object_name

Displays the fully qualified name of the object. The format of this field depends on the object type as shown here:

DATABASE
N/A

DBCAL
<calendar>

DBDOMAIN
<domain>

DBJBSTRM
<workstation>#<job_stream>

DBJOB
 <workstation>#<job>

DBPARM
 <workstation>#<parameter>

DBPROMPT
 <prompt>

DBRES
 <workstation>#<resource>

DBSEC
 N/A

DBUSER
 [<workstation>#]<user>

DBVARTAB
 <variable_table>

DBWKCLS
 <workstation_class>

DBWKSTN
 <workstation>

PLAN N/A

PLDOMAIN
 <domain>

PLFILE
 <workstation>#<path>(<qualifier>)

PLJBSTRM
 <workstation>#<job_stream_instance>

PLJOB
 <workstation>#<job_stream_instance>.<job>

PLPROMPT
 [<workstation>#]<prompt>

PLRES
 <workstation>#<resource>

PLWKSTN
 <workstation>

action_data_fields

Displays the action-specific data fields. The format of this data is dependent on the <action_type> field.

Sample audit log entries

This is a sample database audit log:

```

HEADER |20080207|084124|20080207|094124|DATABASE|      |WK1|      | | |Version=A1.0| Level=1
DATABASE|20080207|084124|20080207|094124|DBRES  |ADD  |WK1|operator1| |res=WK1#RESOURCE  |
DATABASE|20080207|100524|20080207|110524|DBWKSTN |MODIFY|WK1|operator1| |ws=TIVOLI10      |
DATABASE|20080207|100525|20080207|110525|DBWKSTN |MODIFY|WK1|operator1| |ws=ASLUTRI1      |

```

```

DATABASE|20080207|100525|20080207|110525|DBWKSTN |MODIFY|WK1|operator1| |ws=WK1      |
DATABASE|20080207|100526|20080207|110526|DBDOMAIN|MODIFY|WK1|operator1| |dom=MASTERDM|
DATABASE|20080207|100610|20080207|110610|DBWKSTN |MODIFY|WK1|operator1| |ws=TIVOLI10    |
DATABASE|20080207|100610|20080207|110610|DBWKSTN |MODIFY|WK1|operator1| |ws=ASLUTRI1    |
DATABASE|20080207|100611|20080207|110611|DBWKSTN |MODIFY|WK1|operator1| |ws=WK1          |
DATABASE|20080207|100611|20080207|110611|DBWKSTN |ADD   |WK1|operator1| |ws=WK2          |
DATABASE|20080207|100612|20080207|110612|DBDOMAIN|MODIFY|WK1|operator1| |dom=MASTERDM    |

```

This is a sample plan audit log:

```

HEADER  |20080207|100758|20080207|110758|PLAN      |      |WK1|admin| |      |Version=A1.0|Level=1
STAGEMAN|20080207|100758|20080207|110758|PLAN      |INSTALL|WK1|admin| |C:\IBM\TWS\oper1\Symphony|
        AWSBHV030I The new Symphony file is installed.
STAGEMAN|20080207|100758|20080207|110758|PLAN      |INSTALL|WK1|admin| |C:\IBM\TWS\oper1\Sinfonia|
        AWSBHV036I Multi-workstation Symphony file copied to C:\IBM\TWS\oper1\Sinfonia
STAGEMAN|20080207|100758|20080207|110758|ADITLEVL|MODIFY |WK1|admin| |      |
        AWSBHV077I Audit level changing from 0 to 1.
CONMAN  |20080207|100800|20080207|110800|PLWKSTN |MODIFY |      |admin| |WK1
        continue & start
CONMAN  |20080207|100941|20080207|110941|PLWKSTN |MODIFY |      |admin| |SLUTRI1
        limit cpu=slutril;10
PLAN    |20080207|101018|20080207|111018|PLWKSTN |MODIFY |WK1|oper1| |WK1
        limit cpu=SLUTRI1;20
PLAN    |20080207|101028|20080207|111028|PLDOMAIN|MODIFY |WK1|oper1| |ECCOLO
        reply ECCOLO;yes

```

A **ResetPlan** command run against the current production plan is stored in the plan audit log file as follows:

```

STAGEMAN|20080207|100758|20080207|110758|PLAN|DELETE|WK1|admin|
        |/home/WK1/schedlog/M200803140127|
        AWSBHV025I The old Symphony file renamed /home/WK1/schedlog/M200803140127

```

Dynamic workload scheduling audit

Description

When you select the dynamic scheduling capability at installation time, the auditing feature is automatically installed. By default, the auditing feature is disabled.

Auditable events are as follows:

JobDefinitionAuditEvent

Maintains a track of operations performed on job definitions.

JobLogAuditEvent

Maintains a track of operations performed on job logs.

JobAuditEvent

Maintains a track of operations performed on jobs.

ResourceAuditEvent

Maintains a track of operations performed on resources.

RelationshipAuditEvent

Maintains a track of operations performed on relationships between resources.

RecoveryActionAuditEvent

Maintains a track of operations performed on recovery actions.

HistoryDataAuditEvent

Maintains a track of operations performed on historical data.

To configure the auditing of events, enable the auditing feature and optionally change the default values in the configuration file to define event types to be audited. The configuration file is located in the following path:

TWA_home\TDWB\config\audit.properties

Configuring the audit

Configure one or more of the properties in the *audit.properties* file to enable and configure auditing:

audit.enabled

Specifies whether the auditing feature is enabled or disabled. The default value is false. Supported values are as follows:

false The auditing feature is not enabled.

true The auditing feature is enabled.

onSecurityEnabled

The auditing feature is enabled if global security is enabled on the WebSphere Application Server.

audit.consumer.file.auditFilePrefix

Specifies the file prefix for the auditing log file. The file name is defined using the file prefix plus the *_auditN.log* suffix, where *N* is a progressive number. If you want the date and time of the file creation specified in the file prefix, use the default format: 'tdwb_'yyyy-MM-dd. For instance, using the default prefix 'tdwb_'yyyy-MM-dd generates the *tdwb_2010-12-20_auditN.log* family of files. Note that the text between single quotes (') is not processed by the program and remains unchanged. This format creates a different file for each day the auditing feature is enabled. Also, changing the prefix to 'tdwb_'yyyy-MM generates the *tdwb_2010-12_auditN.log* family of files. This format creates a different file for each month the auditing feature is enabled.

You can modify this format as desired to create files on a weekly, monthly or yearly basis, depending on your auditing requirements. Depending on the date and time format you choose, the maximum size and number of log files vary. The maximum size and number of log files are defined using the **audit.consumer.file.maxFileSize** and

audit.consumer.file.maxAuditFiles properties respectively. Use these three parameters to control the size of the audit logs stored. For example, using the default values for these parameters, then every day you will have a maximum of 10 MB x 100 files each day. Once the maximum is reached, the first file created is overwritten. If you want use less space to store audit

logs, you can decide to change the maximum number of files or only have files on a monthly basis, by specifying the format for the `audit.consumer.file.auditFilePrefix` property as `'tdwb_''yyyy-MM`.

audit.consumer.file.auditFileLocation

Specifies the path where the log files are created. The default path is `/audit`.

audit.consumer.file.maxFileSize

Specifies the maximum size in bytes of the log files. When a file reaches the maximum size, a new log file is created. The default value is 10000000 bytes (10 MB). This is also the highest supported value.

audit.consumer.file.maxAuditFiles

Specifies the maximum number of files with a specific prefix. When all files reach the maximum size and the maximum number of files is exceeded, the oldest file with a specific prefix is overwritten. The default value is 100 files. This is also the highest supported value.

Configuring dynamic audit events

The following table lists the supported actions and properties for each event with the related default values. You can configure these values in the `audit.properties` file.

Table 7. Auditable event properties

Event	Action	Property	Default value
JobDefinitionAuditEvent	create	<code>audit.tdwb.JobDefinitionAuditEvent.create.enabled</code>	true
	delete	<code>audit.tdwb.JobDefinitionAuditEvent.delete.enabled</code>	true
	get	<code>audit.tdwb.JobDefinitionAuditEvent.get.enabled</code>	true
	query	<code>audit.tdwb.JobDefinitionAuditEvent.query.enabled</code>	false
	update	<code>audit.tdwb.JobDefinitionAuditEvent.update.enabled</code>	true
JobLogAuditEvent	get	<code>audit.tdwb.JobLogAuditEvent.get.enabled</code>	true
JobAuditEvent	cancel	<code>audit.tdwb.JobAuditEvent.cancel.enabled</code>	true
	get	<code>audit.tdwb.JobAuditEvent.get.enabled</code>	true
	query	<code>audit.tdwb.JobAuditEvent.query.enabled</code>	false
	submit	<code>audit.tdwb.JobAuditEvent.submit.enabled</code>	true
ResourceAuditEvent	create	<code>audit.tdwb.ResourceAuditEvent.create.enabled</code>	true
	delete	<code>audit.tdwb.ResourceAuditEvent.delete.enabled</code>	true
	query	<code>audit.tdwb.ResourceAuditEvent.query.enabled</code>	false
	resume	<code>audit.tdwb.ResourceAuditEvent.resume.enabled</code>	true
	suspend	<code>audit.tdwb.ResourceAuditEvent.suspend.enabled</code>	true
	update	<code>audit.tdwb.ResourceAuditEvent.update.enabled</code>	true
RelationshipAuditEvent	create	<code>audit.tdwb.RelationshipAuditEvent.create.enabled</code>	true
	delete	<code>audit.tdwb.RelationshipAuditEvent.delete.enabled</code>	true
	query	<code>audit.tdwb.RelationshipAuditEvent.query.enabled</code>	false
RecoveryActionAuditEvent	invoke	<code>audit.tdwb.RecoveryActionAuditEvent.invoke.enabled</code>	true
HistoryDataAuditEvent	move	<code>audit.tdwb.HistoryDataAuditEvent.move.enabled</code>	true

By default, auditing is disabled for query actions, while all the other actions are enabled. If the auditing feature is disabled, all properties are ignored.

Log file specifications

The elements used in the auditing log files are extensions to the Common Base Event (CBE) schema. The types and elements listed below are available in the auditing log files. Supported action types for each element are listed in Table 7 on page 77.

Action

Represents the action that is being taken. Each auditable event supports a different set of possible actions. See Table 7 on page 77. The Action type contains the following element:

Table 8. Elements in Action type

Element name	Element description	Always returned in the output
Action	The action type that is being taken on the dynamic workload broker object.	Yes

ObjectInfoList

Represents a list of dynamic workload broker objects. The ObjectInfoList type contains the following element:

Table 9. Elements in ObjectInfoList type

Element name	Element description	Always returned in the output
objectInfo	The class of the object being involved in the action	Yes

ObjectInfo

Represents information about a dynamic workload broker object in an objectInfoList type or in another objectInfo element. The ObjectInfo type contains the following elements:

Table 10. Elements in ObjectInfo type

Element name	Element description	Always returned in the output
objectClass	The class of the object being involved in the action.	Yes
objectName	The name of the dynamic workload broker object.	Only if available
objectNamespace	The namespace of the dynamic workload broker object.	Only if available
objectType	The type of the dynamic workload broker object.	Only if available
objectAlias	The alias of the dynamic workload broker object.	Only if available
objectIdentifier	The unique identifier of the dynamic workload broker object.	Only if available

Table 10. Elements in ObjectInfo type (continued)

Element name	Element description	Always returned in the output
objectRole	The role of the dynamic workload broker object, if any. For instance a Resource can have the source or destination role in a relationship	Only if available
objectSubmitterType	The type of the component which submitted the operation. The component is one of the following: <ul style="list-style-type: none"> • Tivoli Dynamic Workload Broker Console • Command line • Dynamic workload broker workstation • Third party utility 	Only if available
objectInfo	A child objectInfo object. For instance, a relationship is always related to two resources.	Only if available

Outcome

Defines the outcome of a security event. The Outcome type contains the following elements:

Table 11. Elements in Outcome type

Element name	Element description	Always returned in the output
result	The status of the event. This information can be used when filtering the information in the log file.	Yes
failureReason	Additional information on the outcome of the operation.	Yes, if the operation was unsuccessful.

UserInfoList

Represents a list of userInfo elements, each representing the list of users in the delegation chain. The UserInfoList type contains the following element:

Table 12. Elements in UserInfoList type

Element name	Element description	Always returned in the output
objectInfo	An array of Information about each user in the delegation chain. The first userInfo element identifies the user which authenticated first. The last userInfo element identifies the user with whose credentials the action is being taken.	Yes

UserInfo

Represents information about a user. Elements of this type return information about the user involved in the operation being audited. The UserInfo type contains the following element:

Table 13. Elements in UserInfo type

Element name	Element description	Always returned in the output
UserInfo	The username provided to dynamic workload broker for authentication.	Yes

How to perform queries on log files

Log files can be very long and detailed. When you view your log files with the Log and Trace Analyzer, you can apply one or more queries to filter information in the file and make searches faster. You can use the following queries to filter only the relevant information or you can create your own queries depending on your requirements. The following queries are written in XPath query language.

- To filter all the events generated by a specific user:
`/CommonBaseEvent [extendedDataElements/children[@name='userInfo' and values='username']]`
- To filter all the events related to a specific object class:
`/CommonBaseEvent [extendedDataElements/ /children[@name='objectClass' and values='Resource']]`
- To filter all the events related to a specific object:
`//CommonBaseEvent [extendedDataElements/ /children[@name='objectName' and values='myresource']/../children[@name='objectClass' and values='Resource']]`
- To filter all the events related to a specific action:
`/CommonBaseEvent [extendedDataElements[@name='action' and values='uninstall']]`
- To filter all the events with SUCCESSFUL outcome:
`/CommonBaseEvent [extendedDataElements/children[@name='result' and values='SUCCESSFUL']]`

The following query returns all create actions:

```
/CommonBaseEvent[ extendedDataElements[@name = 'action' and values = 'create']]
```

You can export this query into an XML file as follows:


```

<?xml version="1.0" encoding="UTF-8"?><cbeviewer_configuration>
<logParserSets>
  <logParserSet description="Parser for CBE log"
    id="com.ibm.cbeviewer.parsers.cbeLogParserSet"
    label="Common Base Event log"
    parentId="com.ibm.cbeviewer.parsers.jdLogParserSet"/>
  <logParserSet description="Parser for CEI Server"
    id="com.ibm.cbeviewer.parsers.ceiLogParserSet"
    label="Common Event Infrastructure server"
    parentId="com.ibm.cbeviewer.parsers.jdLogParserSet"/>
  <logParserSet description="Other parsers"
    id="com.ibm.cbeviewer.parsers.otherParsersLogParserSet"
    label="Other parsers"/>
</logParserSets>
<recent_expressions>
  <xpath name="All Create Events">
    /CommonBaseEvent[ extendedDataElements[@name = 'action' and values = 'create']]
  </xpath>
</recent_expressions></cbeviewer_configuration>

```

The following is a short example of a log file:

```

<CommonBaseEvent
  creationTime="2007-06-06T14:26:23.311Z"
  extensionName="TDWB_JOB_AUDIT_EVENT"
  globalInstanceId="CEFC6DD156CA54D902A1DC1439E6EC4ED0"
  sequenceNumber="1"
  version="1.0.1">
  <extendedDataElements
    name="userInfoList"
    type="noValue">
    <children
      name="userInfo"
      type="string">
      <values>UNAUTHENTICATED</values>
    </children>
  </extendedDataElements>
  <extendedDataElements
    name="action"
    type="string">
    <values>submit</values>
  </extendedDataElements>
  <extendedDataElements
    name="outcome"
    type="noValue">
    <children
      name="result"
      type="string">
      <values>SUCCESSFUL</values>
    </children>
  </extendedDataElements>
</CommonBaseEvent>

```

Examples

The following examples describe a standard usage of the auditing feature.

In the following example, user root successfully retrieves the definition of a job named **MyTestJob** using the jobstore command.

```

<CommonBaseEvent
  creationTime="2007-06-21T16:05:19.455Z"
  extensionName="TDWB_JOB_AUDIT_EVENT"
  globalInstanceId="CE8F5E102AE3419AF7A1DC201135463A40"
  sequenceNumber="188"
  version="1.0.1">
  <extendedDataElements
    name="userInfoList"

```

```

        type="noValue">
      <children
        name="userInfo"
        type="string">
      <values>root</values>
    </children>
  </extendedDataElements>
  <extendedDataElements
    name="action"
    type="string">
  <values>get</values>
</extendedDataElements>
<extendedDataElements
  name="outcome"
  type="noValue">
  <children
    name="result"
    type="string">
  <values>SUCCESSFUL</values>
  </children>
</extendedDataElements>
<extendedDataElements
  name="objectInfoList"
  type="noValue">
  <children
    name="objectInfo"
    type="noValue">
    <children
      name="objectClass"
      type="string">
    <values>Job</values>
    </children>
    <children
      name="objectName"
      type="string">
    <values>MyTestJob</values>
    </children>
    <children
      name="objectIdentifier"
      type="string">
    <values>3ebf6d62-0b83-3270-9b83-83c393e9cbca</values>
    </children>
    <children
      name="objectSubmitterType"
      type="string">
    <values>TDWB CLI</values>
    </children>
    </children>
  </extendedDataElements>
  <extendedDataElements
    name="CommonBaseEventLogRecord:sequenceNumber"
    type="long">
  <values>80808</values>
</extendedDataElements>
<extendedDataElements
  name="CommonBaseEventLogRecord:threadID"
  type="int">
  <values>280</values>
</extendedDataElements>
<sourceComponentId
  application="JobManagement"
  component="None"
  componentIdType="Application"
  location="tdws08"
  locationType="Hostname"
  subComponent="None"
  threadId="Default : 84"
  componentType="http://www.ibm.com/namespace/autonomic/Tivoli_componentTypes"/>
<situation
  categoryName="ReportSituation">
  <situationType

```

```

        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="ReportSituation"
        reasoningScope="INTERNAL"
        reportCategory="SECURITY"/>
    </situation>
</CommonBaseEvent>

```

In the following example, user `testuser` tries deleting a job instance named **MySecondJob** using the appropriate command line. The operation fails because the job was submitted by another user. Deleting jobs submitted by other users requires Operator or Administrator rights. For more information on access rights, see *IBM Tivoli Workload Scheduler: Scheduling Workload Dynamically* or *IBM Tivoli Workload Scheduler: Administration Guide*.

```

<CommonBaseEvent
  creationTime="2007-06-21T16:05:32.746Z"
  extensionName="TDWB_JOB_AUDIT_EVENT"
  globalInstanceId="CE8F5E102AE3419AF7A1DC20113D32BB20"
  sequenceNumber="189"
  version="1.0.1">
  <extendedDataElements
    name="userInfoList"
    type="noValue">
    <children
      name="userInfo"
      type="string">
      <values>testuser</values>
    </children>
  </extendedDataElements>
  <extendedDataElements
    name="action"
    type="string">
    <values>cancel</values>
  </extendedDataElements>
  <extendedDataElements
    name="outcome"
    type="noValue">
    <children
      name="result"
      type="string">
      <values>UNSUCCESSFUL</values>
    </children>
    <children
      name="failureReason"
      type="string">
      <values>userNotAuthorized</values>
    </children>
  </extendedDataElements>
  <extendedDataElements
    name="objectInfoList"
    type="noValue">
    <children
      name="objectInfo"
      type="noValue">
      <children
        name="objectClass"
        type="string">
        <values>Job</values>
      </children>
      <children
        name="objectName"
        type="string">
        <values>MySecondJob</values>
      </children>
      <children
        name="objectIdentifier"
        type="string">
        <values>a05732c8-c008-3103-afd1-84b567d78de7</values>
      </children>
    </children>
  </extendedDataElements>

```

```

        name="objectSubmitterType"
        type="string">
        <values>TDWB CLI</values>
    </children>
</extendedDataElements>
<extendedDataElements
    name="CommonBaseEventLogRecord:sequenceNumber"
    type="long">
    <values>80964</values>
</extendedDataElements>
<extendedDataElements
    name="CommonBaseEventLogRecord:threadID"
    type="int">
    <values>292</values>
</extendedDataElements>
<sourceComponentId
    application="JobManagement"
    component="None"
    componentIdType="Application"
    location="tdws08"
    locationType="Hostname"
    subComponent="None"
    threadId="Default : 91"
    componentType="http://www.ibm.com/namespace/autonomic/Tivoli_componentTypes"/>
<situation
    categoryName="ReportSituation">
    <situationType
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="ReportSituation"
        reasoningScope="INTERNAL"
        reportCategory="SECURITY"/>
    </situation>
</CommonBaseEvent>

```

Chapter 6. Troubleshooting performance issues

This refers you to the *Administration Guide* for the resolution of performance problems.

The performance of Tivoli Workload Scheduler can depend on many factors. Preventing performance problems is at least as important as resolving problems that occur. For this reason, all discussion of performance issues has been placed together in the chapter on performance in the *Tivoli Workload Scheduler: Administration Guide*.

Chapter 7. Troubleshooting networks

Describes how to recover from short-term and long-term network outages and offers solutions to a series of network problems.

This section describes how to resolve problems in the Tivoli Workload Scheduler network. It covers the following topics:

- “Network recovery”
- “Other common network problems” on page 89

Network recovery

Several types of problems might make it necessary to follow network recovery procedures. These include:

- Initialization problems that prevent agents and domain managers from starting properly at the start of a new production period. See “Initialization problems.”
- Network link problems that prevent agents from communicating with their domain managers. See “Network link problems” on page 88.
- Loss of a domain manager, which requires switching to a backup. See “Replacement of a domain manager” on page 89.
- Loss of a master domain manager, which is more serious, and requires switching to a backup or other more involved recovery steps. See “Replacement of a master domain manager” on page 89.

Note: In all cases, a problem with a domain manager affects all of its agents and subordinate domain managers.

Initialization problems

Initialization problems can occur when Tivoli Workload Scheduler is started for a new production period. This can be caused by having Tivoli Workload Scheduler processes running on an agent or domain manager from the previous production period or a previous Tivoli Workload Scheduler run. To initialize the agent or domain manager in this situation, do the following:

1. For a domain manager, log into the parent domain manager or the master domain manager. For an agent, log into the agent domain manager, the parent domain manager, or the master domain manager.
2. Run the Console Manager and issue a **stop** command for the affected agent.
3. Run a **link** command for the affected agent. This initializes and starts the agent.

If these actions fail to work, check to see if netman is running on the affected agent. If not, issue the **StartUp** command locally and then issue a **link** command from its domain manager.

If there are severe network problems preventing the normal distribution of the new Symphony file, a fault-tolerant agent or subordinate domain manager can be run as a standalone system, provided the following conditions are met:

- The Sinfonia file was generated on the master domain manager after the network problem occurred, and so has never been transferred to the agent or domain manager

- You have some other method, such as a physical file transfer or FTP to transfer the new Sinfonia file from the master domain manager to the agent or subordinate domain manager
- The master domain manager and the agent or subordinate domain manager have the same processor architecture

If these conditions are met, do the following

1. Stop the agent or domain manager
2. Delete the `<TWA_home>/TWS/Symphony` file on the agent or domain manager
3. Copy the file `<TWA_home>/TWS/Sinfonia` from the master domain manager to the `<TWA_home>/TWS` directory on the agent or domain manager.
4. Rename the copied file `<TWA_home>/TWS/Symphony`
5. Run **StartUp** to start the agent or domain manager.

Any inter-workstation dependencies must be resolved locally using appropriate console manager commands, such as **Delete Dependency** and **Release**.

Network link problems

Tivoli Workload Scheduler has a high degree of fault tolerance in the event of a communications problem. Each fault-tolerant agent has its own copy of the Symphony file, containing the production period's processing. When link failures occur, they continue processing using their own copies of the Symphony file. Any inter-workstation dependencies, however, must be resolved locally using appropriate console manager commands: **deldep** and **release**, for example.

While a link is down, any messages destined for a non-communicating workstations are stored by the sending workstations in the `<TWA_home>/TWS/pobox` directory, in files named `<workstation>.msg`. When the links are restored, the workstations begin sending their stored messages. If the links to a domain manager are down for an extended period of time, it might be necessary to switch to a backup (see *Tivoli Workload Scheduler: Administration Guide*).

Note:

1. The **conman submit job** and **submit schedule** commands can be issued on an agent that cannot communicate with its domain manager, provided that you configure (and they can make) a direct HTTP connection to the master domain manager. This is configured using the *conman connection* options in the `localopts` file, or the corresponding options in the `useropts` file for the user (see the *Tivoli Workload Scheduler: Administration Guide* for details).
However, all events have to pass through the domain manager, so although jobs and job streams can be submitted, their progress can only be monitored locally, not at the master domain manager. It is thus always important to attempt to correct the link problem as soon as possible.
2. If the link to a standard agent workstation is lost, there is no temporary recovery option available, because standard agents are hosted by their domain managers. In networks with a large number of standard agents, you can choose to switch to a backup domain manager.

Troubleshooting a network link problem

When an agent link fails it is important to know if the problem is caused by your network or by Tivoli Workload Scheduler. The following procedure is run from the master domain manager to help you to determine which:

1. Try using **telnet** to access the agent: **telnet <node>:<port>**
2. Try using **ping** to access the agent: **ping <node>:<port>**

3. Run **nslookup** for the agent and the master domain manager from both, and check that the information on each system is the same from each system
4. Run **netstat -a |grep <port>** and check if any FIN_WAIT_2 states exist
5. Verify that the port number of the master domain manager matches the entry for "nm port" in the localopts file of the master domain manager
6. Verify that the port number of the agent matches the entry for "nm port" in the localopts file of the agent
7. Check the netman and TWSMerge logs on both the master domain manager and the agent, for errors.

Note:

1. Any issues found in steps 1 on page 88 to 4 suggest that there are problems with the network
2. Any issues found in steps 5 to 7 suggest that there are problems with the Tivoli Workload Scheduler configuration or installation

If this information does not provide the answer to the linking issue, call IBM Software Support for further assistance.

The commands used in steps 1 on page 88 to 4 are IP network management commands, information about which can be obtained in the Internet. The following technical note also provides useful information about their use:
<http://www.ibm.com/support/docview.wss?rs=0&uid=swg21156106>

Replacement of a domain manager

A domain manager might need to be changed as the result of network linking problems or the failure of the domain manager workstation itself. It can be temporarily replaced by switching any full status agent in its domain to become the new domain manager, while the failed domain manager is repaired or replaced.

The steps for performing this activity are as described for the planned replacement of a domain manager; see *Tivoli Workload Scheduler: Administration Guide*.

Replacement of a master domain manager

If you lose a master domain manager, you have to perform all of the steps described in *Tivoli Workload Scheduler: Administration Guide* for the planned replacement of a master domain manager.

Other common network problems

The following problems could be encountered:

- "Using SSL, no connection between a fault-tolerant agent and its domain manager" on page 90
- "After changing SSL mode, a workstation cannot link" on page 90
- "In a configuration with a firewall, the start and stop remote commands do not work" on page 91
- "The domain manager cannot link to a fault-tolerant agent" on page 91
- "Changes to the SSL keystore password prevent the application server from starting" on page 92
- "Agents not linking to master domain manager after first JnextPlan on HP-UX" on page 92

- “Fault-tolerant agents not linking to master domain manager” on page 93
- “The dynamic agent cannot be found from Dynamic Workload Console” on page 94
- “Submitted job is not running on a dynamic agent” on page 94
- “Job status of a submitted job is continually shown as running on dynamic agent” on page 94
- “Network performance” on page 95
- “AWSITA245E or AWSITA104E - Agent is down but jobmanager is running” on page 95

Using SSL, no connection between a fault-tolerant agent and its domain manager

In a network using SSL authentication, no connection can be established between a fault-tolerant agent and its domain manager. The standard lists of the two workstations display messages like the following:

- On the domain manager, mailman messages:

```
+ ++++++
+ AWSBCV082I Workstation FTAHP, Message: AWSDEB009E Data
+ transmission is not possible because the connection is broken.
+ The following gives more details of the error: Error 0.
+ ++++++
+ AWSBCV035W Mailman was unable to link to workstation: rsmith297;
+ the messages are written to the PO box.
+ ++++++
```

- On the fault-tolerant agent, writer messages:

```
/* *****
/* AWSBCW003E Writer cannot connect to the remote mailman. The
/* following gives more details of the error: "
/* AWSDEB046E An error has occurred during the SSL handshaking. The
/* following gives more details of the error: error:140890B2:SSL
/* routines:SSL3_GET_CLIENT_CERTIFICATE:no certificate returned
/* *****
/* AWSDEZ003E **ERROR**(cpu secs 0)
```

Cause and solution:

In the localopts file of either the domain manager or the fault-tolerant agent , the SSL port statement is set to 0.

Correct the problem by setting the SSL port number to the correct value in the localopts file. You then need to stop and restart netman on the workstation so that it can now listen on the correct port number

After changing SSL mode, a workstation cannot link

You have changed the SSL mode between a workstation and its domain manager. However, you are unable to relink to the workstation from the domain manager.

Cause and solution:

The Symphony file and message files at the workstation must be deleted after a change of SSL mode, otherwise the data does not match. The files to delete are the following:

```
Symphony
Sinfonia
$HOME/*.msg
$HOME/pobox/*.msg
```

In a configuration with a firewall, the start and stop remote commands do not work

In a configuration with a firewall between the master domain manager and one or more domain managers, the **start** and **stop** commands from the master domain manager to the fault-tolerant agents in the domains do not work. This is often the case when an "rs final" ends and the impacted fault-tolerant agents are not linked.

Cause and solution:

The fault-tolerant agents belonging to these domains do not have the *behind firewall* attribute set to *on* in the Tivoli Workload Scheduler database. When there is a firewall between the master domain manager and other domains, **start** and **stop** commands must go through the Tivoli Workload Scheduler hierarchy. This parameter tells the master domain manager that the **stop** request must be sent to the domain manager which then sends it to the fault-tolerant agents in its domain.

Use either the Dynamic Workload Console or the composer **cpuname** command to set to the *behind firewall* attribute *on* in the workstation definitions of these fault-tolerant agents.

Remote command job fails to connect to remote computer

After submitting a remote command job, error message AWKRCE012E indicates that an error has occurred establishing a connection to the remote computer.

Cause and solution:

There are several possible causes to this problem:

- The host name specified for the computer where the remote command instance is running does not exist.
- The port number is incorrect, for example, a port number which is different from the port number configured for a specific protocol.
- The protocol type specified is unable to establish a connection because the remote computer is not open to using that particular protocol.

See Job definition - remote command jobs for information about the connection settings for the remote computer.

The domain manager cannot link to a fault-tolerant agent

The domain manager cannot link to a fault-tolerant agent. The stdlist records the following messages:

```
+ ++++++
+ AWSEDW020E: Error opening IPC
+ AWSEDW001I: Getting a new socket: 9
+ ++++++
```

Cause and solution:

The fault-tolerant agent has two netman processes listening on the same port number. This is the case if you installed more than one Tivoli Workload Scheduler instance on the same workstation and failed to specify different netman port numbers.

Stop one of the two netman services and specify a unique port number using the nm port local option (localopts file).

Ensure that the workstation definition on the master domain manager is defined with the unique port number or it will not be able to connect.

Changes to the SSL keystore password prevent the application server from starting

You change the password to the SSL keystore on the application server, or you change the security settings using the WebSphere Application Server **changeSecuritySettings** tool. The application server does not start. The following message is found in the application server's trace file trace.log (the message is shown here on three lines to make it more readable):

```
JSAS0011E: [SSLConfiguration.validateSSLConfig] Java. exception
Exception = java.io.IOException:
Keystore was tampered with, or password was incorrect
```

This problem is discussed in "The application server does not start after changes to the SSL keystore password" on page 122.

Agents not linking to master domain manager after first JnextPlan on HP-UX

You have successfully installed the components of your network with the master domain manager on HP-UX. You perform all the necessary steps to create a plan and run your first **JnextPlan**, which appears to work correctly. The Symphony file is distributed to the agents but they cannot link to the master domain manager, even if you issue a specific **link** command for them. The **conman** error log shows that the agents cannot communicate with the master domain manager.

Cause and solution:

One possible cause for this problem is that while on HP-UX host names are normally limited to eight bytes, on some versions of this platform you can define larger host names. The problem occurs if you define the master domain manager's host name as more than eight bytes. When you install the master domain manager on this host a standard operating system routine obtains the host name from the operating system, but either truncates it to eight bytes before storing it in the database, or stores it as "unknown". When you install the agents, you supply the longer master domain manager host name. However, when the agents try to link to the master domain manager they cannot match the host name.

To resolve this problem, do the following:

1. Change the workstation definition of the master domain manager to the correct host name
2. Run **ResetPlan -scratch**
3. Run **JnextPlan**

The agents now link.

Fault-tolerant agents not linking to master domain manager

A fault-tolerant agent does not link to its master domain manager and any other link problem scenarios documented here do not apply.

Cause and solution:

The cause of this problem might not be easy to discover, but is almost certainly involved with a mismatch between the levels of the various files used on the fault-tolerant agent.

To resolve the problem, if all other attempts have failed, perform the following cleanup procedure. However, note that this procedure loses data (unless the fault-tolerant agent is not linking after a fresh installation), so should not be undertaken lightly.

Do the following:

1. Using **conman "unlink @;noask"** or the Dynamic Workload Console, unlink the agent from the master domain manager
2. Stop Tivoli Workload Scheduler, in particular netman, as follows:
 - a. **conman "stop;wait"**
 - b. **conman "shut;wait"**
 - c. On Windows only; **shutdown**
 - d. Stop the SSM agent, as follows:
 - On Windows, stop the Windows service: Tivoli Workload Scheduler SSM Agent (for <TWS_user>).
 - On UNIX, run **stopmon**.

Note: If the **conman** commands do not work, try the following:

UNIX `ps -ef | grep <TWS_user> & kill -9`

Windows

`<TWA_home>\TWS\unsupported\listproc & killproc`

3. **Risk of data loss:** Removing the following indicated files can cause significant loss of data. Further, if jobs have run on the fault-tolerant agent for the current plan, without additional interaction, the fault-tolerant agent will rerun those jobs.

Remove or rename the following files:

```
<TWS_home>\TWS\*.msg
          \Symphony
          \Sinfonia
          \Jobtable
          \pobox\*.msg
```

Note: See Chapter 14, "Corrupt Symphony file recovery," on page 205 for additional options.

4. Start **netman** with **StartUp** run as the TWS_user

5. Issue a **"link"** command from the master domain manager to the fault-tolerant agent
6. Issue a **conman start** command on the fault-tolerant agent.

The IBM technical note describing this procedure also contains some advice about starting with a lossless version of this procedure (by omitting step 3 on page 93) and then looping through the procedure in increasingly more-aggressive ways, with the intention of minimizing data loss. See <http://www.ibm.com/support/docview.wss?uid=swg21296908>

The dynamic agent cannot be found from Dynamic Workload Console

You correctly installed a dynamic agent but cannot see it from the Dynamic Workload Console.

Cause and solution:

A possible cause for this problem might be that either the dynamic workload broker hostname, **-tdwbhostname**, or the dynamic workload broker port, or both, and which are both registered on the agent, are not known in the network of the master domain manager because the broker host is in a different DNS domain.

Edit the JobManager.ini configuration file (for its path, see "Where products and components are installed" on page 1). Edit the following parameter:

```
ResourceAdvisorUrl = https://<servername>:  
31116/JobManagerRESTWeb/JobScheduler/resource
```

Submitted job is not running on a dynamic agent

From Dynamic Workload Console, you can see a dynamic agent, but the submitted job appears as "No resources available" or is dispatched to other agents.

Cause and solution:

A possible cause might be that the local hostname of a registered dynamic workload broker server on the agent is not known in the network of the master domain manager because it is in a different DNS domain.

Edit the JobManager.ini configuration file (for its path, see "Where products and components are installed" on page 1). Edit the following parameter:

```
FullyQualifiedHostname = <servername>
```

Job status of a submitted job is continually shown as running on dynamic agent

From Dynamic Workload Console, you can see a dynamic agent, but the job status of a submitted job is continually in the running state.

Cause and solution:

A possible cause might be that the master domain manager local hostname is not known in the network of the agent because it is in a different DNS domain.

Open the JobDispatcherConfig.properties file and edit the parameter
JDURL=https://<localhostname>

See the *Administration Guide* for more details about editing this file.

Network performance

If your network shows performance problems, you can use the environment variable `TWS_TRUSTED_ADDRESS` to resolve these problems. You can set `TWS_TRUSTED_ADDRESS` to an IP address or leave it blank. If you set it to an IP address, all the processes use this address for their connections. If you leave it blank, the processes use the address returned by the operating system. If you do not create this variable, the processes use the IP address returned by the Symphony file.

AWSITA245E or AWSITA104E - Agent is down but jobmanager is running

If while using **conman** or the Dynamic Workload Console you receive one of the following error messages stating that the agent is down, while jobmanager process is up and running, causes and solutions might be as follows:

Cause and solution

AWSITA245E

An error occurred getting the response of the HTTP request.

- If the detailed error reports problems about CURL Error ##, check the following link: <http://curl.haxx.se/libcurl/c/libcurl-errors.html>.
- If the error is 35, perform the following steps:
 1. Enable **gskit** traces, as described in the following section: <http://www-01.ibm.com/support/docview.wss?uid=swg21283690>.
 2. Edit the `JobManager.ini` file and add the following line
`GSK_TRACE_FILE = /tmp/gskit.log` in the ITA Envsection.
 3. Reproduce the problem.
 4. Collect the `/tmp/gskit.log` file.

AWSITA104E

Unable to perform the system resources scan. The problem is that the *Common Inventory Technology* (CIT) is unable to perform correctly the resources scan because the hostname of the machine is not recognized. The CIT return codes are documented at the following site: <http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/index.jsp?topic=%2Fcom.ibm.tivoli.itcm.doc%2FInvug265.htm>

To solve the problem, perform the following steps:

1. Run the `/opt/tivoli/bin/cit/wscanhw` command and check if you obtain the same result.
2. Enable the traces of the JobManager process, wait for the error to occur and check the `JobManager_trace.log` file.
 - If the error is executed with error(s)! exit why is <1>, exit code is <1>, check the command line used by running the following and parameters:

```
/home/twsuser/TWS/bin/citLauncher --cit -task WSCANHW -c /home/twsuser/TWS/ITA/cpa/ita/fullScan.xml
```
 - Run the `/opt/tivoli/bin/cit/wscanhw` command again, as twsuser.

- |
 - |
 - |
3. If it still fails, check that the permissions of the `/home/twsuser/TWS/bin/citLauncher` file are: `-r-s--s--x 1 root tws86`.

Chapter 8. Troubleshooting common engine problems

Gives solutions to problems that might occur with the modules and programs that comprise the basic scheduling "engine" on the master domain manager.

This section details commonly occurring problems and their solutions in components and activities not already discussed in previous chapters.

Other common problems are dealt with in other guides, or other chapters of this guide:

- For installation problems see the *Tivoli Workload Scheduler: Planning and Installation Guide*.
- For network problems see Chapter 7, "Troubleshooting networks," on page 87
- For problems with the fault-tolerant switch manager see Chapter 12, "Troubleshooting the fault-tolerant switch manager," on page 195.
- For problems with the Symphony file see Chapter 14, "Corrupt Symphony file recovery," on page 205.

The problems are grouped according to their typology:

Composer problems

The following problems could be encountered with composer:

- "Composer gives a dependency error with interdependent object definitions"
- "The display cpu=@ command does not work on UNIX" on page 98
- "Composer gives the error "user is not authorized to access server"" on page 98
- "The deletion of a workstation fails with the "AWSJOM179E error" on page 99
- "When using the composer add and replace commands, a Job Scheduler has synchronicity problems" on page 99

Composer gives a dependency error with interdependent object definitions

You are running composer to add or modify a set of object definitions where one object is dependent on another in the same definition. An error is given for the dependency, even though the syntax of the definition is correct.

Cause and solution:

Composer validates objects in the order that they are presented in the command or the definition file. For example, you define two jobs, and the first-defined (*job_tom*) has a follows dependency on the second-defined (*job_harry*). The object validation tries to validate the follows dependency in *job_tom* but cannot find *job_harry* so gives an error and does not add the job to the database. However, it then reads the definition of *job_harry*, which is perfectly valid, and adds that to the database.

Similarly, this problem could arise if you define that a job needs a given resource or a Job Scheduler needs a given calendar, but you define the resource or calendar *after* defining the job or Job Scheduler that references them.

This problem applies to all composer commands that create or modify object definitions.

To resolve the problem, you can just simply repeat the operation. In the above example the following happens:

- The first job defined (*job_tom*) now finds the second job (*job_harry*) which was added to the database initially.
- You receive a "duplicate job" error for the second.

Alternatively, you can edit the object definition and retry the operation with just the object definition that gave the error initially.

To ensure that the problem does not reoccur, always remember to define objects in the order they are to be used. Define depending jobs and job streams before dependent ones. Define referred objects before referring objects.

Note: There is a special case of this error which impacts the use of the **validate** operation. Because **validate** does not add *any* job definitions to the database, correct or otherwise, *all* interdependent job definitions give an error.

In the example above, the problem would not have occurred when using **add**, **new**, **create**, or **modify** if the job definition of *job_harry* preceded that of *job_tom*. *job_harry* would have been added to the database, so the validation of *job_tom* would have been able to verify the existence of *job_harry*. Because the **validate** command does not add *job_harry* to the database, the validation of the follows dependency in *job_tom* fails.

There is no workaround for this problem when using **validate**. All you can do is to ensure that there are no interdependencies between objects in the object definition file.

The **display cpu=@** command does not work on UNIX

In UNIX, nothing happens when typing **display cpu=@** at the composer prompt.

Cause and solution:

The @ (atsign) key is set up as the "kill" character.

Type **stty -a** at the UNIX prompt to determine the setting of the @ key. If it is set as the "kill" character, then use the following command to change the setting to be "control/U" or something else:

```
stty kill ^U
```

where ^U is "control/U", not caret U.

Composer gives the error "user is not authorized to access server"

Troubleshooting for a user authorization error in composer.

You successfully launch composer but when you try to run a command, the following error is given:

```
user is not authorized to access server
```

Cause and solution:

This is a problem that is common to several CLI programs; see “Command line programs (like composer) give the error “user is not authorized to access server”” on page 141.

The deletion of a workstation fails with the "AWSJOM179E error

You want to delete a workstation either using **Composer** or the Dynamic Workload Console and the following error occurs:

AWSJOM179E An error occurred deleting definition of the workstation {0}.
The workload broker server is currently unreachable.

Cause and solution:

This problem occurs if you removed a dynamic domain manager without following the procedure that describes how to uninstall a dynamic domain manager in the *Tivoli Workload Scheduler: Planning and Installation Guide*.

To remove workstations connected to the dynamic domain manager, perform the following steps:

1. Verify that the dynamic domain manager was deleted, not just unavailable, otherwise when the dynamic domain manager restarts, you must wait until the workstations register again on the master domain manager before using them.
2. Delete the workstations using the following command:

```
composer del ws <workstation_name>;force
```

When using the composer add and replace commands, a Job Scheduler has synchronicity problems

The composer **add** and **replace** commands do not correctly validate the time zone used in the Job Scheduler definition at daylight savings; as a consequence, the following unexpected warning message is displayed:

```
AWSBIA148W  WARNING: UNTIL time occurs before AT time for  
            <workstation>#<schedule>.  
AWSBIA019E  For <workstation>#<schedule> Errors 0, warnings 1.  
AWSBIA106W  The schedule definition has warnings.  
AWSBIA015I  Schedule <workstation>#<schedule> added.
```

The same might happen for the deadline keyword.

Cause and solution:

The problem is related to the C-Runtime Library date and time functions that fail to calculate the correct time during the first week of daylight savings time.

To ensure the accuracy of scheduling times, for the time argument of the `at`, `until`, or `deadline` scheduling keywords, specify a different value than that of the start time for the Tivoli Workload Scheduler production period defined in the global options file. These values must differ from one another by plus or minus one hour.

JnextPlan problems

The following problems could be encountered with JnextPlan:

- “JnextPlan fails to start” on page 100

- "JnextPlan fails with the database message "The transaction log for the database is full.""
- "JnextPlan fails with a Java out-of-memory error" on page 101
- "JnextPlan fails with the DB2 error like: nullDSRA0010E" on page 101
- "JnextPlan fails with message AWSJPL017E" on page 102
- "JnextPlan is slow" on page 102
- "A remote workstation does not initialize after JnextPlan" on page 102
- "A job remains in "exec" status after JnextPlan but is not running" on page 103
- "A change in a resource quantity in the database is not also implemented in the plan after JnextPlan" on page 104
- "On SLES8, after the second JnextPlan, an agent does not link" on page 105
- "CreatePostReports.cmd, or Makeplan.cmd, or Updatestats.cmd hang on Windows operating systems" on page 105

JnextPlan fails to start

JnextPlan fails to start.

Cause and solution:

This error might be a symptom that your Tivoli Workload Scheduler network requires additional tuning because of a problem with the sizing of the pobox files. The default size of the pobox files is 10MB. You might want to increase the size according to the following criteria:

- The role (master domain manager, domain manager, or fault-tolerant agent) of the workstation in the network. Higher hierarchical roles need larger pobox files due to the larger number of events they must handle (since the total number of events that a workstation receives is proportional to the number of its connections). For a domain manager, also the number of sub domains under its control make a difference.
- The average number of jobs in the plan.
- The I/O speed of the workstation (Tivoli Workload Scheduler is IO- dependent).

JnextPlan fails with the database message "The transaction log for the database is full."

You receive a message from **JnextPlan** which includes the following database message (the example is from DB2, but the Oracle message is very similar):

The transaction log for the database is full.

The **JnextPlan** message is probably the general database access error message AWSJDB801E.

Cause and solution:

The problem is probably caused by the number of Job Scheduler instances that **JnextPlan** needs to handle. The default database transaction log files cannot handle more than the transactions generated by a certain number of Job Scheduler instances. In the case of DB2 this number is 180 000; in the case of Oracle it depends on how you configured the database. If **JnextPlan** is generating this many instances, you need to change the log file creation parameters to ensure more log space is created. You might also need to increase the Java heap size on the

application server. See "Scalability" in the *Tivoli Workload Scheduler: Administration Guide* for a full description of how to perform these activities.

JnextPlan fails with a Java out-of-memory error

You receive the following message from **JnextPlan**:

```
AWSJCS011E An internal error has occurred.  
The error is the following: "java.lang.OutOfMemoryError".
```

Cause and solution:

This error is a symptom that the processes running in the WebSphere Application Server during **JnextPlan** phase need more Java virtual memory to run. To increase the Java virtual memory for the WebSphere Application Server, you must increase the default Java heap size values.

In the file

```
<WAS_profile_path>/config/cells/  
TWSNodeCell/nodes/TWSNode/servers/server1/server.xml
```

where the default path for <WAS_profile_path> is *TWA_home*/WAS/TWSPprofile, modify the values of the `initialHeapSize` and `maximumHeapSize` variables as described in the Heap size settings for jvm 64 bits table.

Note: The minimum value for `initialHeapSize` is 512 MB and for `maximumHeapSize` is 1024 MB.

For more information about increasing the application server heap size, see the *Tivoli Workload Scheduler Administration Guide: Scalability*.

JnextPlan fails with the DB2 error like: nullDSRA0010E

JnextPlan has failed with the following messages:

```
AWSJPL705E An internal error has occurred. The planner is unable to create  
the preproduction plan.
```

```
AWSBIS348E An internal error has occurred. MakePlan failed while running:  
planman.
```

```
AWSBIS335E JnextPlan failed while running: tc1sh84
```

The `SystemOut.log` has an error like this:

```
AWSJDB801E An internal error has been found while accessing the database.  
The internal error message is: "nullDSRA0010E: SQL State = 57011, Error  
Code = -912".
```

Cause and solution:

This indicates that the memory that DB2 allocates for its "lock list" is insufficient. To understand why the problem has occurred and resolve it, see the section in the *Tivoli Workload Scheduler: Administration Guide* about monitoring the "lock list" value among the DB2 administrative tasks.

JnextPlan fails with message AWSJPL017E

You receive the following message from **JnextPlan**:

AWSJPL017E The production plan cannot be created because a previous action on the production plan did not complete successfully. See the message help for more details.

Cause and solution:

The problem might be caused by a **JnextPlan** being launched before the previous **JnextPlan** has run the **SwitchPlan** command.

The situation might not resolve itself. To resolve it yourself, do the following:

1. Reset the plan by issuing the command **ResetPlan -scratch**
2. If the reset of the plan shows that the database is locked, run a **planman unlock** command.

JnextPlan is slow

You find that JnextPlan is unacceptably slow.

Cause and solution:

There are three possible causes for this problem:

Tracing too much

One possible cause is the tracing facility. It could be that it is providing too much trace information. There are three possible solutions:

- Reduce the number of processes that the tracing facility is monitoring. See "Quick reference: how to modify log and trace levels" on page 9 for full details.
- Stop the tracing facility while JnextPlan is running. To do this issue the following command before it starts:

```
atctl off TWS all
```

Issue the following command to switch the tracing back on again:

```
atctl on TWS all
```

This can be automated within a script that launches JnextPlan.

Application server tracing too much

Another possible cause is that the application server tracing is set to high. See "Log and trace files for the application server" on page 37 for more details about the trace and how to reset it.

Database needs reorganizing

Another possible cause is that the database needs reorganizing. See "Reorganizing the database" in *Tivoli Workload Scheduler: Administration Guide* for a description of how and why you reorganize the database, logically and physically.

A remote workstation does not initialize after JnextPlan

After running JnextPlan you notice that a remote workstation does not immediately initialize. The following message is seen:

```

+++++
+ AWSBCW037E Writer cannot initialize this workstation because mailman
+ is still active.
+ +++++
+ AWSBCW039E Writer encountered an error opening the Mailbox.msg file.
+ The total cpu time used is as follows: 0
+++++

```

Cause and solution:

If mailman is still running a process on the remote workstation, JnextPlan cannot download the Symphony file and initialize the next production period's activities. Instead, the domain manager issues a **stop** command to the workstation. The workstation reacts in the normal way to the **stop** command, completing those activities it must complete and stopping those activities it can stop.

After the interval determined in the `localopts` parameter *mm retrylink*, the domain manager tries again to initialize the workstation. When it finds that the **stop** command has been implemented, it starts to initialize the workstation, downloading the Symphony file and starting the workstation's activities.

A workstation does not link after JnextPlan

On UNIX and Windows operating systems, after running JnextPlan a workstation does not link to the domain manager.

Cause and solution:

The workstation does not recognize its own IP address and host name. To resolve this problem, add the workstation IP address and hostname to the hosts file located in `C:\Windows\System32\Drivers\etc\hosts` (Windows) or `/etc/hosts` (UNIX) file in the following format:

```
IPaddress hostname hostname.domain
```

For example, `9.168.60.9 nc060009 nc060009.romelab.it.ibm.com`

A job remains in "exec" status after JnextPlan but is not running

After running JnextPlan you notice that a job has remained in "exec" status, but is not being processed.

Cause and solution:

This error scenario is possible if a job completes its processing at a fault-tolerant agent just before JnextPlan is run. The detail of the circumstances in which the error occurs is as follows:

1. A job completes processing
2. The fault-tolerant agent marks the job as "succ" in its current Symphony file
3. The fault-tolerant agent prepares and sends a job status changed event (JS) and a job termination event (JT), informing the master domain manager of the successful end of job
4. At this point JnextPlan is started on the master domain manager

5. **JnextPlan** starts by unlinking its workstations, including the one that has just sent the JS and JT events. The message is thus not received, and waits in a message queue at an intermediate node in the network.
6. **JnextPlan** carries the job forward into the next Symphony file, and marks it as "exec", because the last information it had received from the workstation was the Launch Job Event (BL).
7. **JnextPlan** relinks the workstation
8. The fault-tolerant agent receives the new Symphony file and checks for jobs in the "exec" status.
9. It then correlates these jobs with running processes but does not make a match, so does not update the job status
10. The master domain manager receives the Completed Job Event that was waiting in the network and marks the carried forward job as "succ" and so does not send any further messages in respect of the job
11. Next time **JnextPlan** is run, the job will be treated as completed and will not figure in any further Symphony files, so the situation will be resolved. However, in the meantime, any dependent jobs will not have been run. If you are running **JnextPlan** with an extended frequency (for example once per month), this might be a serious problem.

There are two possible solutions:

Leave JnextPlan to resolve the problem

If there are no jobs dependent on this one, leave the situation to be resolved by the next **JnextPlan**.

Change the job status locally to "succ"

Change the job status as follows:

1. Check the job's stdlist file on the fault-tolerant agent to confirm that it did complete successfully.
2. Issue the following command on the fault-tolerant agent:

```
conman "confirm <job>;succ"
```

To prevent the reoccurrence of this problem, take the following steps:

1. Edit the **JnextPlan** script
2. Locate the following command:

```
conman "stop @!@;wait ;noask"
```
3. Replace this command with individual stop commands for each workstation (conman "stop <workstation> ;wait ;noask") starting with the farthest distant nodes in the workstation and following with their parents, and so on, ending up with the master domain manager last. Thus, in a workstation at any level, a message placed in its forwarding queue either by its own job monitoring processes or by a communication from a lower level should have time to be forwarded at least to the level above before the workstation itself is closed down.
4. Save the modified **JnextPlan**.

A change in a resource quantity in the database is not also implemented in the plan after JnextPlan

You make changes to the number of available resources in the database, but the number of available resources in the plan does not change. The global option `enCFResourceQuantity` is set to *no*.

Cause and solution:

If the global option `enCFResourceQuantity` is set to *yes*, you would expect that any changes to the available quantity of a given resource in the database would not be implemented in the plan, provided there is at least one job or Job Scheduler instance using that resource in the extended plan.

Similarly, if the global option `enCFResourceQuantity` is set to *no* you might expect that the available resource quantity would change after **JnextPlan**. However, this is not always true, depending on the quantity of that resource being used by jobs and Job Scheduler instances currently in the plan:

- If the usage of the resource by jobs and Job Scheduler instances is *less than or equal to* the new total of available resources in the database, the available quantity of the resource is changed in the plan.
- If the usage of the resource by jobs and Job Scheduler instances is *greater than* the new total of available resources in the database, the available quantity of the resource is *not* changed in the plan.

To be sure to update the quantity of resources in the plan, make available at least as many instances of the resource as are required by the jobs and Job Scheduler instances in the plan.

See also the description of the `enCFResourceQuantity` option in the *Tivoli Workload Scheduler: User's Guide and Reference*.

On SLES8, after the second JnextPlan, an agent does not link

You have installed an agent on SLES8. The first **JnextPlan** works fine, but the second fails, with **conman** giving an error.

Cause and solution:

The problem is caused by a missing library on the agent workstation, called `ligcc_s.so.1`.

The **conman** process cannot run without this library, and **JnextPlan** uses **conman** to stop Tivoli Workload Scheduler processes that were started after the Symphony file arrived after the first **JnextPlan**. That is why **JnextPlan** did not fail the first time, because **JnextPlan** detected that processes were not running and did not need to use **conman** to stop them.

This is a library that is normally in `/lib`, but in this case is not. Look for it in other directories, such as `/usr/lib`. If you cannot locate it on your computer, contact IBM Software Support for assistance.

When you have located it, make a soft link to it from the `/lib` directory and rerun **JnextPlan**.

CreatePostReports.cmd, or Makeplan.cmd, or Updatestats.cmd hang on Windows operating systems

CreatePostReports.cmd, or Makeplan.cmd, or Updatestats.cmd hang on Windows operating systems

Cause and solution:

On Windows operating systems it might happen that, when running `CreatePostReports.cmd`, or `Makeplan.cmd`, or `Updatestats.cmd` jobs, the Tool Command Language interpreter hangs not returning an answer to the caller and the jobs do not complete. These jobs use by default the Tool Command Language interpreter but can be configured to not use it. To avoid that `CreatePostReports.cmd`, or `Makeplan.cmd` or `Updatestats.cmd` jobs use the Tool Command Language interpreter, open each of them with a text editor, find the following line:

```
set USETCL=yes
```

and change it either commenting it:

```
REM set USETCL=yes
```

or setting the variable to no:

```
set USETCL=no
```

In this way, when `CreatePostReports.cmd`, or `Makeplan.cmd` or `Updatestats.cmd` jobs run, the Tool Command Language interpreter is not invoked.

Conman problems

The following problems could be encountered when running **conman**:

- “On Windows, the message AWSDEQ024E is received”
- “Conman on a SLES8 agent fails because a library is missing” on page 107
- “Duplicate ad-hoc prompt number” on page 108
- “Submit job streams with a wildcard loses dependencies” on page 108

On Windows, the message AWSDEQ024E is received

When attempting to log in to **conman** on a Windows operating system, the following error is received:

```
+++++
+ AWSDEQ024E Error owner is not of type user in TOKENUTILS.C;1178
+++++
```

Cause and solution:

This problem can have a variety of causes related to users and permissions. Check the following on the server:

<TWS_user> **password**

Make sure that the password that you supplied for the <TWS_user> user is correct, that the account is not locked out, and that the password has not expired.

Tokenstrv service

Ensure that the Tivoli Token Service (tokenstrv) is started by the Tivoli Workload Scheduler administrative user (not the local system account). This must be verified in the properties of that service in the Services panel; see *Tivoli Workload Scheduler: Administration Guide* for details of how to access that panel and view the details of the user that "owns" the service.

If the password to this user has changed on the workstation, check also that the password has been changed in the entry on the Services panel.

File ownerships

Check that the following ownerships are correct:

- All .exe and .dll files in the <TWA_home>\TWS\bin directory are owned by the <TWS_user>
- All .cmd files are owned by "Administrator"

If necessary, alter the ownership of these files as follows:

1. Stop any active Tivoli Workload Scheduler processes.
2. Change to the <TWA_home>\TWS directory.
3. Issue the following commands:

```
setown -u <TWS_user> .\bin\*.exe
setown -u <TWS_user> .\bin\*.dll c:\win32app\maestro>
setown -u administrator .\bin\*.cmd
```
4. Issue a **StartUp** command on the affected server.
5. On the Tivoli Workload Scheduler master domain manager, launch **conman**.
6. Once conman is started, issue the following command sequence: **link @!@;noask**
7. Keep issuing the **sc** command to ensure that all the servers relink. A server is considered linked if the State shows "LTI JW"

Advanced user rights

Make sure that the <TWS_user> has the correct advanced user rights, as documented in the *Tivoli Workload Scheduler: Planning and Installation Guide*. These are as follows:

- Act as part of the operating system
- Adjust memory quotas for a process
- Log on as a batch job
- Log on as a service
- Log on locally
- Replace a process level token
- Impersonate a client after authentication right

Resolving the problem by reinstalling

If none of the above suggestions resolve the problem, you might need to reinstall Tivoli Workload Scheduler. However, it might happen that the uninstallation fails to completely remove all of the Registry keys from the previous installation. In this case, remove the registry keys following the procedure in the *Tivoli Workload Scheduler: Planning and Installation Guide*. Then make a fresh installation from the product DVD, subsequently reapplying the most recent fix pack, if there is any.

Conman on a SLES8 agent fails because a library is missing

You are running **conman** on an agent on Linux SLES8. A message is received indicating that **conman** cannot be run because the library `libgcc_s.so.1` is missing.

Cause and solution:

This is a library that is normally in `/lib`, but in this case is not. Look for it in other directories, such as `/usr/lib`. If you cannot locate it on your computer, contact IBM Software Support for assistance.

When you have located it, make a soft link to it from the `/lib` directory and rerun **JnextPlan**.

Duplicate ad-hoc prompt number

You issue a job or Job Scheduler that is dependent on an ad-hoc prompt, but **conman** cannot submit the job because the prompt number is duplicated.

Cause and solution:

On the master domain manager, prompts are created in the plan using a unique prompt number. This number is maintained in the file of the master domain manager. **JnextPlan** initially sets the prompt number to "1", and then increments it for each prompt that is to be included in the plan.

If you want to submit a job or Job Scheduler using an ad-hoc prompt on another Tivoli Workload Scheduler agent during the currency of a plan, the local **conman** looks in its own `runmsgno` file in its own `<TWA_home>/TWS/mozart/` directory, and uses the number it finds there. The value in the local file does not necessarily reflect the current value used in the Symphony file. For example, when the file is first created on an agent the run number is created as the highest run number used in the Symphony file at that time, plus 1000. It is then incremented every time **conman** needs to assign a number to a prompt. Despite this interval of 1000, it is still possible for duplicates to occur.

To resolve the problem, edit the file and change the number. An example of the file contents is as follows:

```
0      1236
```

The format is as follows:

- The 10-digit last Symphony run number, right-justified, blank filled. This should not be edited.
- A single blank
- The 10-digit last prompt number, right-justified, blank filled.

For example:

```
123456789012345678901
0      98
```

When modifying the last prompt number, remember that the least significant digit must always be in character position 21. This means that if the current number is "98" and you want to modify it to display "2098" then you must replace two spaces with the "20", and not just insert the two characters. For example:

```
123456789012345678901
0      2098
```

Save the file and rerun the submit. No error should be given by **conman**.

Submit job streams with a wildcard loses dependencies

You issue a submit of interdependent job streams using a wildcard. In certain circumstances you lose the dependencies in an anomalous way.

Cause and solution::

To understand the cause, follow this example, in which the job streams are represented by A, B, C, and their instances are represented by 1, 2:

1. You have the following job streams and jobs in the Symphony file:

A1
B1 (A1,C1)
C1

where B1 depends on A1 and C1.

2. You submit all the jobs, using:

sbs @

The planner creates the following Job Scheduler instances:

A2
B2 (A2,C1)
C2

B2 now depends on A2 and C1. This is correct, because at the moment of submitting the B2 Job Scheduler C2 did not exist, so the highest instance available was C1.

3. The planner then asks you to confirm that you want to submit the instances:

Do you want to submit A2?
Do you want to submit B2?
Do you want to submit C2?

4. Assume that you do not want to submit the job streams A2 and C2, yet, so you reply "No" to the first and last questions. In these circumstances you lose the dependency on A2, but not on C1. This behavior is correct and logical but could be seen by some as anomalous.

To correct the situation, stop the agent on the workstation where the Job Scheduler is running and cancel the Job Scheduler. Then determine the correct sequence of actions to perform to achieve your desired objective and submit the appropriate jobs.

Fault-tolerant agent problems

The following problems could be encountered with fault-tolerant agents.

- "A job fails in heavy workload conditions"
- "Batchman, and other processes fail on a fault-tolerant agent with the message AWSDEC002E" on page 110
- "Fault-tolerant agents unlink from mailman on a domain manager" on page 110

A job fails in heavy workload conditions

A job fails on a fault-tolerant agent where a large number of jobs are running concurrently and one of the following messages is logged:

- "TOS error: No space left on device."
- "TOS error: Interrupted system call."

Cause and solution:

This problem could indicate that one or more of the CCLog properties has been inadvertently set back to the default values applied in a prior version (which used to occasionally impact performance).

See "Tivoli Workload Scheduler logging and tracing using CCLog" on page 14 and check that the TWSCCLog.properties file contains the indicated default values for the properties twsHnd.logFile.className and twsloggers.className.

If the correct default values are being used, contact IBM Software Support to address this problem.

Batchman, and other processes fail on a fault-tolerant agent with the message AWSDEC002E

The batchman process fails together with all other processes that are running on the fault-tolerant agent, typically mailman and jobman (and JOBMON on Windows 2000). The following errors are recorded in the stdlist log of the fault-tolerant agent:

```
+ ++++++
+ AWSBCV012E Mailman cannot read a message in a message file.
+ The following gives more details of the error:
+ AWSDEC002E An internal error has occurred. The following UNIX
+ system error occurred on an events file: "9" at line = 2212
+ ++++++
```

Cause and solution:

The cause is a corruption of the file Mailbox.msg, probably because the file is not large enough for the number of messages that needed to be written to it.

Consider if it seems likely that the problem is caused by the file overflowing:

- If you are sure that this is the cause, you can delete the corrupted message file.

All events lost: Following this procedure means that all events in the corrupted message file are lost.

Perform the following steps:

1. Use the **evtsize** command to increase the Mailbox.msg file. Ensure that the file system has sufficient space to accommodate the larger file.
 2. Delete the corrupt message file.
 3. Restart Tivoli Workload Scheduler by issuing the conman **start** command on the fault-tolerant agent.
- If you do not think that this is the answer, or are not sure, contact IBM Software Support for assistance.

Fault-tolerant agents unlink from mailman on a domain manager

A message is received in the maestro log on the domain manager from mailman for each of the fault-tolerant agents to which it is connected. The messages are as follows:

```
MAILMAN:06:15/ + ++++++
MAILMAN:06:15/ + WARNING: No incoming from <<workstation>>
                - disconnecting. [2073.25]
MAILMAN:06:15/ + ++++++
```

These messages usually occur in the 30 - 60 minutes immediately following JnextPlan.

Cause and solution:

This problem is normally caused by a false timeout in one of the mailman processes on the domain manager. During the initialization period immediately

following JnextPlan, the "*.msg" files on the domain manager might become filled with a backlog of messages coming from fault-tolerant agents. While mailman is processing the messages for one fault-tolerant agent, messages from other fault-tolerant agents are kept waiting until the configured time interval for communications from a fault-tolerant agent is exceeded, at which point mailman unlinks them.

To correct the problem, increase the value of the *mm response* and *mm unlink* variables in the configuration file `~maestro/localopts`. These values must be increased together in small increments (60-300 seconds) until the time-outs no longer occur.

Dynamic agent problems

The following problems could be encountered with dynamic agent.

- "The dynamic agent cannot contact the server"
- "V8.5.1 fault-tolerant agent with dynamic capabilities cannot be registered"
- "Error message AWKDBE009E is received" on page 112

The dynamic agent cannot contact the server

The dynamic agent cannot communicate with the server.

The dynamic agent cannot contact the Tivoli Workload Scheduler master domain manager or dynamic domain manager.

Cause and solution:

This problem might indicate that the list of URLs for connecting to the master domain manager or dynamic domain manager stored on the dynamic agent is incorrect. Perform the following steps:

1. Stop the dynamic agent
2. Delete the `BackupResourceAdvisorUrls` property from the `JobManager.ini` file
3. Edit the `ResourceAdvisorUrl` property in the `JobManager.ini` file and set the URL of the master domain manager or dynamic domain manager.
4. Start the dynamic agent.

V8.5.1 fault-tolerant agent with dynamic capabilities cannot be registered

Describes how to resolve the problem of a V8.5.1 fault-tolerant agent that cannot be registered with its master domain manager.

You have installed a fault-tolerant agent with dynamic capabilities, version 8.5.1, in a domain controlled by a version 8.6 master domain manager. When you try and register the agent manually with the master domain manager, and the name you want to give the agent is its hostname, an error is given because an agent with that name already exists.

Cause and solution:

This problem is caused because the V8.5.1 fault-tolerant agent with dynamic capabilities is actually two agents, the fault-tolerant agent and a lightweight dynamic agent. This dynamic agent registers itself automatically with the master

domain manager using the hostname as its registered name. When you try and register the fault-tolerant agent manually, an error is given.

To solve this problem, you can perform one of the following operations:

- Give the fault-tolerant agent a name other than the hostname
- Rename the dynamic agent

To avoid encountering this problem in the future, register the agent using its hostname before installing it. When the dynamic agent attempts to register itself automatically, it discovers that an agent with its hostname already exists and registers itself as `<hostname>_1` (or `<hostname>_2`, `<hostname>_3` and so on.)

This is not a problem with V8.6 dynamic agents, because you cannot register a fault-tolerant agent. using its hostname.

Error message AWKDBE009E is received

Submission of an MSSQL job or of a Database job on an MSSQL database fails.

When you try to submit an MSSQL job or a Database job running on an MSSQL database, an error message similar to the following is returned, despite the required JDBC driver being installed in the correct directory:

```
AWKDBE009E Unable to create the connection - "  
java.lang.UnsupportedOperationException:  
Java Runtime Environment (JRE) version 1.6 is not supported by this driver.  
Use the sqljdbc4.jar class library, which provides support for JDBC 4.0."
```

Cause and solution:

Verify that only the required **sqljdbc4.jar** driver is present in the JDBC driver directory. If unsupported JDBC drivers are also present in this directory, the dynamic agent might load them and cause the error message.

To solve the problem, perform the following steps:

1. Remove the unsupported JDBC drivers.
2. Stop the dynamic agent with command **ShutDownLwa**.
3. Restart the dynamic agent with command **StartUpLwa**.

For more information, see the section about configuring to schedule job types with advanced options in *Tivoli Workload Scheduler: Administration Guide*.

Error message AWSITA104E is received

System resources scan fails.

From the **Tivoli dynamic workload broker > Tracking > Computers** entry of the Dynamic Workload Console, if you see the *status* of the agent *online*, but the *availability* is *unavailable*, you can see in the `JobManager_trace.log` file the following error:

```
AWSITA104E Unable to perform the system resources scan.  
The error is "Unable to parse hardware scan output".  
AWSITA105E Unable to notify scan results to the server  
because of a resources scanner error.
```

Cause and solution:

The problem is that the *Common Inventory Technology* (CIT) is unable to perform correctly the resources scan because the hostname of the machine is not recognized.

To solve the problem, perform the following steps:

on UNIX operating systems

- verify that the hostname is listed in the `/etc/hosts` file.
- verify that the `ping hostname` command is performed successfully.

on Windows operating systems

- verify that the `ping hostname` command is performed successfully.

Problems on Windows

You could encounter the following problems running Tivoli Workload Scheduler on Windows.

- “Interactive jobs are not interactive using Terminal Services”
- “The Tivoli Workload Scheduler services fail to start after a restart of the workstation”
- “The Tivoli Workload Scheduler for user service (batchup) fails to start” on page 114
- “An error relating to impersonation level is received” on page 115

Interactive jobs are not interactive using Terminal Services

You want to run a job at a Windows fault-tolerant agent, launching the job remotely from another workstation. You want to use Windows Terminal Services to launch the job on the fault-tolerant agent, either with the Dynamic Workload Console or from the command line. You set the “is interactive” flag to supply some run time data to the job, and indicate the application program that is to be run (for example, `notepad.exe`). However, when the job starts running, although everything seems correct, the application program window does not open on the Terminal Services screen. An investigation at the fault-tolerant agent shows that the application program is running on the fault-tolerant agent, but Terminal Services is not showing you the window.

Cause and solution:

The problem is a limitation of Terminal Services, and there is no known workaround. All “interactive jobs” must be run by a user at the fault-tolerant agent, and cannot be run remotely, using Terminal Services. Jobs that do not require user interaction are not impacted, and can be run from Terminal Services without any problems.

The Tivoli Workload Scheduler services fail to start after a restart of the workstation

On Windows, both the Tivoli Token service and the Tivoli Workload Scheduler for user service (batchup) fail to start after a restart of the workstation on which they are running.

Cause and solution:

The user under which these services start might have changed password.

If you believe this to be the case, follow the procedure described in *Tivoli Workload Scheduler: Administration Guide*.

The Tivoli Workload Scheduler for user service (batchup) fails to start

The Tivoli Workload Scheduler for `<TWS_user>` service (sometimes also called *batchup*) does not start when the other Tivoli Workload Scheduler processes (for example, mailman and batchman) start on workstations running Windows 2000 and 2003 Server. This problem occurs on a fault-tolerant agent, either after a `conman` start command or after a domain manager switch. The Tivoli Token service and netman services are unaffected.

This problem does not impact scheduling, but can result in misleading status data.

Cause and solution:

The problem is probably caused either because the `<TWS_user>` has changed password, or because the name of the service does not match that expected by Tivoli Workload Scheduler. This could be because a change in the configuration of the workstation has impacted the name of the service.

To resolve the problem temporarily, start the service manually using the Windows Services panel (under **Administrative Tools**). The service starts and runs correctly. However, the problem could reoccur unless you correct the root cause.

To resolve the problem permanently, follow these steps:

1. If the `<TWS_user>` has changed password, ensure that the service has been changed to reflect the new password, as described in *Tivoli Workload Scheduler: Administration Guide*.
2. Look at the Windows Event Viewer to see if the information there explains why the service did not start. Resolve any problem that you find.
3. If the reason given for the failure of the service to start is the following, this normally means that there is a mismatch between the name of the installed service, and the name of the service that the mailman process calls when it starts:

System error code 1060:
The specified service does not exist as an installed service

The normal reason for this is that the user ID of the `<TWS_user>` has changed. The `<TWS_user>` cannot normally be changed by you, so this implies some change that has been imposed externally. A typical example of this is if you have promoted the workstation from *member server* to *domain controller*. When this happens, the local `<TWS_user>` is converted automatically to a domain user, which means that the domain name is prefixed to the user ID, as follows: `<domain_name>\<TWS_user>`.

The problem occurs because of the way Tivoli Workload Scheduler installs the service. If the workstation is *not* a domain controller the installation names the service: `twc_maestro_<TWS_user>`. If the workstation *is* a domain controller the installation names the service: `twc_maestro_<domain_name>_<TWS_user>`.

When batchman starts up it discovers that the `<TWS_user>` is a domain user. Batchman tries to use the domain user service name to start the batchup service. The action fails because the service on the workstation has the local user service name.

To resolve this problem you must change the name of this service, and to do this you are recommended to uninstall the Tivoli Workload Scheduler instance and re-install it.

An alternative, but deprecated, method is to change the name of the service in the Windows registry.

Attention: Making changes to the Windows Registry can make the operating system unusable. You are strongly advised to back up the Registry before you start.

If you decide to use this method you must edit the following keys:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\tws_maestro_<TWS_user>
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\tws_maestro_<TWS_user>
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet002\Services\tws_maestro_<TWS_user>
```

They must be changed to the following:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\tws_maestro_<domain_name>_<TWS_user>
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\tws_maestro_<domain_name>_<TWS_user>
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet002\Services\tws_maestro_<domain_name>_<TWS_user>
```

If you have changed the name of the service in the registry, you must ensure that the logon is correct. Open the **Log On** tab of the service in the Windows Services panel and change the account name, if necessary, to `<domain_name>\<TWS_user>`. You must also enter the password and confirm it.

An error relating to impersonation level is received

On Windows, an error is received when you try to use any of the Tivoli Workload Scheduler commands (for example, **conman**, **composer**, **datecalc**). The error message is similar to the following:

```
AWSDEQ008E Error opening thread token ../../src/libs/tokenutils.c:1380
message = Either a required impersonation level was not provided, or the
provided impersonation level is invalid
```

Cause and solution:

This issue occurs when the user account that is used to run the Tivoli Workload Scheduler command line does not have the user right: "Impersonate a client after authentication". This is a new security setting that was first introduced in the following service packs:

Windows 2000

Service Pack 4

Windows XP

Service Pack 2

Windows 2003

All versions.

Windows 7

Windows 2008

The upgrade does not grant this right to existing users.

For full details of this right, see the appropriate Windows publication.

To resolve this problem, grant the user right "Impersonate a client after authentication" to all users that need to run Tivoli Workload Scheduler commands on the workstation. To do this, follow these steps:

1. Select **Start** → **Programs** → **Administrative Tools** → **Local Security Policy**
2. Expand **Local Policies**, and then click **User Rights Assignment**.
3. In the right pane, double-click **Impersonate a client after authentication**.
4. In the Local Security Policy Setting dialog box, click **Add**.
5. In the Select Users or Group dialog box, click the user account that you want to add, click **Add**, and then click **OK**.
6. Click **OK**.

Extended agent problems

The following problem could be encountered with extended agents:

The return code from an extended agent job is not recognized

You have a network including Tivoli Workload Scheduler versions 8.5, 8.4, 8.3, 8.2, or 8.2.1 and Tivoli Workload Scheduler for Applications, version 8.1.1. An extended agent job (submitted either through the Dynamic Workload Console or **conman**), has given an unrecognized return code.

Cause and solution:

If Tivoli Workload Scheduler does not receive a return code from the extended agent job, it substitutes the return code with the exit code of the method. If this last is zero, the job has finished successfully. If it is not zero, contact IBM Software Support for an explanation of the exit code and a resolution of the problem.

Planner problems

The following problems could be encountered with the planner:

There is a mismatch between Job Scheduler instances in the Symphony file and the preproduction plan

You notice that there are Job Scheduler instances in the Symphony file that are not in the preproduction plan.

Cause and solution:

Job streams are automatically deleted from the preproduction plan when they are completed. However, it is possible to set the "carryStates" global option (using **optman**) so that job streams with jobs in the *SUCC* status are carried forward. In this case such job streams are carried forward to the new Symphony file when the plan is extended, but are deleted from the preproduction plan if the job streams have been successfully completed. This is not an error. These job streams can remain in the current plan (Symphony file) and can even be run again.

To resolve the situation for a given plan, use **conman** or the Dynamic Workload Console to delete the Job Scheduler instances from the plan.

To prevent the problem reoccurring, consider why the "carryStates" global option is set so that job streams with jobs in the *SUCC* status are carried forward. If it has been set in error, or is no longer required, change the settings of the option (using **optman**) so that this no longer happens.

Planman deploy error when deploying a plug-in

When using the planman deploy command to deploy a plug-in, the deploy fails with the following error:

```
AWSJCS011E An internal error has occurred. The error is the following:  
"ACTEX0019E The following errors  
from the Java compiler cannot be parsed:  
error: error reading <file_name>; Error opening zip file  
<file_name>
```

Cause and solution:

The .jar file identified in the message is corrupt. Check and correct the format of the file before retrying the deploy.

An insufficient space error occurs while deploying rules

When using the planman deploy command with the -scratch option to deploy all non-draft rules, the following error occurs:

```
AWSJCS011E An internal error has occurred. The error is the following:  
"ACTEX0023E The Active Correlation Technology compiler cannot  
communicate with the external Java compiler.  
java.io.IOException: Not enough space".
```

Cause and solution:

This error occurs when there is insufficient swap space (virtual memory) to perform the operation.

Create more swap space or wait until there are fewer active processes before retrying the operation.

UpdateStats fails if it runs more than two hours (message AWSJCO084E given)

When running the **UpdateStats** command in a large plan, if the job run time exceeds two hours, the job fails, with messages that include the following:

```
AWSJCO084E The user "UNAUTHENTICATED" is not authorized to work with the  
"planner" process.
```

Cause and solution:

This error occurs because the large number of jobs in the plan has caused the job run time to exceed two hours, which is the default timeout for the user credentials of the WebSphere Application Server.

To increase the timeout so that the **UpdateStats** command has more time to run, perform the following:

1. Locate the following file:
`<WAS_profile_path>/config/cells/
TWSNodeCell/nodes/TWSNode/servers/server1/server.xml`

where the default path for `<WAS_profile_path>` is `<TWA_home>/WAS/TWSprofile`.

2. Locate the parameter: `authValidationConfig="system.LTPA" timeout="120"`
3. Edit the timeout value from 120 minutes to a value you think will be sufficient.
4. Stop and restart the WebSphere Application Server, using the `conman stopappserver` and `startappserver` commands (or, in the latter case, the `StartUp` command).

The planman showinfo command displays inconsistent times

The plan time displayed by the `planman showinfo` command might be incongruent with the time set in the operating system of the workstation. For example, the time zone set for the workstation is GMT+2 but `planman showinfo` displays plan times according to the GMT+1 time zone.

Cause and solution:

This situation arises when the WebSphere Application Server Java virtual machine does not recognize the time zone set on the operating system.

As a workaround for this problem, set the time zone defined in the `server.xml` file equal to the time zone defined for the workstation in the Tivoli Workload Scheduler database. Proceed as follows:

1. Stop WebSphere Application Server
2. Create a backup copy of the following file:
`<WAS_profile_path>/config/cells/
TWSNodeCell/nodes/TWSNode/servers/server1/server.xml`

where the default path for `<WAS_profile_path>` is `TWA_home/WAS/TWSprofile`.

3. Open the original file with a text or XML editor
4. Find the `genericJvmArguments` string and add:
`genericJvmArguments="-Duser.timezone=time_zone"`

where `time_zone` is the time zone defined for the workstation in the Tivoli Workload Scheduler database.

5. Restart WebSphere Application Server

A bound z/OS shadow job is carried forward indefinitely

A z/OS shadow job, defined for the distributed environment, is successfully bound to a remote z/OS job, but the z/OS shadow job never completes and is carried forward indefinitely.

Cause and solution:

A Refresh Current Plan operation was performed on the remote Tivoli Workload Scheduler for z/OS instance. Because this operation scratches the current plan, the remote z/OS job instance binding was removed.

To prevent the z/OS shadow job from being indefinitely carried forward, manually cancel the z/OS shadow job instance in the distributed engine plan.

For instructions on how to do this, see the *cancel job* topic in the *Managing objects in the plan - conman* chapter of the *Tivoli Workload Scheduler: User's Guide and Reference*.

Problems with DB2

The following problems could be encountered with DB2:

- "Timeout occurs with DB2"
- "JnextPlan fails with the DB2 message "The transaction log for the database is full.""
- "The DB2 UpdateStats job fails after 2 hours"
- "DB2 might lock while making schedule changes" on page 120

Timeout occurs with DB2

You are trying to edit an object, but after a delay an error is given by DB2 referring to a timeout, similar to the following:

```
AWSJDB803E
```

```
An internal deadlock or timeout error has occurred while processing a  
database transaction. The internal error message is:
```

```
"The current transaction has been rolled back because of a deadlock or timeout.  
Reason code "68".
```

Cause and solution:

In this case the object you are trying to access is locked by another user, or by you in another session, but the lock has not been detected by the application. So the application waits to get access until it is interrupted by the DB2 timeout.

By default, both DB2 and WebSphere Application Server have the same length timeout, but as the WebSphere Application Server action starts before the DB2 action, it is normally the WebSphere Application Server timeout that is logged:

```
AWSJC0005E WebSphere Application Server has given the following error:
```

```
CORBA NO_RESPONSE 0x4942fb01 Maybe; nested exception is:
```

```
org.omg.CORBA.NO_RESPONSE:
```

```
Request 1685 timed out vmcid:
```

```
IBM minor code: B01 completed: Maybe.
```

To resolve the problem, check if the object in question is locked. If it is, take the appropriate action to unlock it, working with the user who locked it. If it is not locked retry the operation. If the problem persists contact IBM Software Support for assistance.

JnextPlan fails with the DB2 message "The transaction log for the database is full."

You receive a message from **JnextPlan** which includes the following DB2 message:
The transaction log for the database is full.

The **JnextPlan** message is probably the general database access error message AWSJDB801E.

Cause and solution:

This scenario is described in "JnextPlan fails with the database message "The transaction log for the database is full." on page 100.

The DB2 UpdateStats job fails after 2 hours

You are running the DB2 **UpdateStats** job, but after 2 hours it fails. The log contains messages similar to the following:


```

[2/20/08 8:22:11:947 CET] 0000001e ServiceLogger I
    com.ibm.ws.ffdc.IncidentStreamImpl initialize FFDC0009I:
    FFDC opened incident stream file /opt/ibm/TWA0/WAS/TWSprofile/
    logs/ffdc/server1_78387838_08.02.20_08.22.11_0.txt
[2/20/08 8:22:11:957 CET] 0000001e ServiceLogger I
    com.ibm.ws.ffdc.IncidentStreamImpl resetIncidentStream FFDC0010I:
    FFDC closed incident stream file /opt/ibm/TWA0/WAS/TWSprofile/
    /logs/ffdc/server1_78387838_08.02.20_08.22.11_0.txt
[2/20/08 8:22:11:999 CET] 0000001e ConnException E
    com.ibm.tws.conn.exception.ConnSecurityException
    ConnException(String currentMessageID, Object[] currentArgs)
    AWSJC0084E The user "UNAUTHENTICATED" is not authorized to work with
    the "planner" process. UNAUTHENTICATED
[2/20/08 8:22:12:004 CET] 0000001e ConnException E
    com.ibm.tws.conn.exception.ConnException
    ConnException(TWSException e)
    AWSJC0084E The user "UNAUTHENTICATED" is not authorized to work with
    the "planner" process.
[2/20/08 8:22:12:088 CET] 0000001e ExceptionHelp E
    com.ibm.tws.cli.exception.ExceptionHelper
    handleException(Throwable e, String commandName,
    TWSServletResponse response)
    AWSJCL054E The command "LOGREPORT" has failed, for the following reason:
    "AWSJC0084E The user "UNAUTHENTICATED" is not authorized to work with
    the "planner" process.".
    LOGREPORT AWSJC0084E The user "UNAUTHENTICATED" is not authorized to work
    with the "planner" process.
[2/20/08 8:22:12:091 CET] 0000001e ThreadMonitor W
    WSVR0606W: Thread "WebContainer : 2" (0000001e) was previously reported
    to be hung but has completed. It was active for approximately 7200340
    milliseconds. There is/are 0 thread(s) in total in the server that
    still may be hung.

```

Cause and solution:

The problem is with the WebSphere Application Server which has a default authentication timeout of 2 hours. The **UpdateStats** job runs without any interrupt that would allow the WebSphere Application Server to reset its timeout.

To resolve the problem, reset the timeout as follows:

1. Edit the following file with a text editor:

```

<WAS_profile_path>/config/cells/
TWSNodeCell/nodes/TWSNode/servers/server1/server.xml

```

where the default path for <WAS_profile_path> is *TWA_home*/WAS/TWSprofile.

2. Locate the key: `authValidationConfig="system.LTPA" timeout="120"`
3. Change the value of the timeout to an appropriately higher figure (the log of **UpdateStats** shows you how much progress the job had made when it stopped; it should be possible to extrapolate from that how much additional time is required).
4. Save the file.
5. Stop and restart the application server using the **stopappserver** and **startappserver** commands.
6. Rerun **UpdateStats**.

DB2 might lock while making schedule changes

Multiple concurrent changes (modify, delete or create) to job streams or domains might cause a logical deadlock between one or more database transactions. This is a remote but possible problem you might encounter.

This deadlock might take place even if the objects being worked on are different (for example, different job streams).

The problem affects database elements (rows or tables), not Tivoli Workload Scheduler objects, so it is unrelated with the Locked By property of Tivoli Workload Scheduler objects.

The same problem might arise when making concurrent changes for plan generation.

When the deadlock occurs, DB2 rolls back one of the deadlocking threads and the following error is logged in the SystemOut.log of WebSphere Application Server:

```
AWSJDB803E An internal deadlock or timeout error has occurred
while processing a database transaction. The internal error
message is: "The current transaction has been rolled back
because of a deadlock or timeout. Reason code "2"."
```

In general, this type of error is timing-dependent, and the transactions involved must overlap in very specific conditions to generate a deadlock. However it might easily occur during plan generation (either forecast, trial, or current), when the plan includes many objects and DB2 must automatically escalate locks from row to table level, as the number of locked objects exceeds the current maximum limit.

You can mitigate the error by increasing the maximum number of locks that DB2 can hold. Refer to the DB2 Information Center to learn more about the DB2 lock escalation mechanism and to find how to increase the maximum number of concurrent locks.

In the above scenarios, if an interactive user session is rolled back, the user gets an error message but is allowed to repeat the task. Instead, if a script session is rolled back (for example, a script that generates a forecast plan or updates a job stream definition), the script ends in failure.

Problems with Oracle

The following problems could be encountered with Oracle:

- “JnextPlan fails with the database message "The transaction log for the database is full."”
- “You cannot do Oracle maintenance on UNIX after installation” on page 122

JnextPlan fails with the database message "The transaction log for the database is full."

You receive a message from **JnextPlan** which includes a database message similar to the following:

```
The transaction log for the database is full.
```

The **JnextPlan** message is probably the general database access error message AWSJDB801E.

Cause and solution:

This scenario is described in “JnextPlan fails with the database message "The transaction log for the database is full."” on page 100.

You cannot do Oracle maintenance on UNIX after installation

You have installed Tivoli Workload Scheduler, creating the installation directory with the default root user permission. When you switch to the Oracle administration user and try and use the Oracle tools, you encounter access problems.

Cause and solution:

The problem could be that the Oracle administration user does not have "read" permission for the entire path of the Tivoli Workload Scheduler installation directory. For example, if you have created the Tivoli Workload Scheduler installation directory as /opt/myProducts/TWS, the Oracle administration user must have "read" permission for /opt and /myProducts, as well as /TWS.

Give the Oracle administration user read permission for the full path of the Tivoli Workload Scheduler installation directory.

Application server problems

The following problems might occur:

- "Timeout occurs with the application server"
- "The application server does not start after changes to the SSL keystore password"

The application server does not start after changes to the SSL keystore password

You change the password to the SSL keystore on the application server, or you change the security settings using the WebSphere Application Server **changeSecuritySettings** tool. The application server does not start. The following message is found in the application server's trace file trace.log (the message is shown here on three lines to make it more readable):

```
JSAS0011E: [SSLConfiguration.validateSSLConfig] Java. exception  
Exception = java.io.IOException:  
Keystore was tampered with, or password was incorrect
```

Cause and solution:

The certificate has not been reloaded or regenerated. Any change to the keystore password on the server or connector requires the SSL certificate to be reloaded or regenerated to work correctly.

Reload or regenerate the certificate and restart the application server.

To regenerate the certificate issue this command:

```
openssl genrsa -des3 -passout pass:<your_password> -out client.key 2048
```

If you do not want to supply the password openly in the command, omit it, and you will be prompted for it.

Timeout occurs with the application server

You are trying to edit an object, but after a delay an error is given by the WebSphere Application Server referring to a timeout, similar to the following:

```
AWSJC0005E WebSphere Application Server has given the following error:
CORBA NO_RESPONSE 0x4942fb01 Maybe; nested exception is:
  org.omg.CORBA.NO_RESPONSE:
    Request 1685 timed out vmcid:
      IBM minor code: B01 completed: Maybe.
```

Cause and solution:

In this case the object you are trying to access is locked from outside Tivoli Workload Scheduler, maybe by the database administrator or an automatic database function. So the application waits to get access until it is interrupted by the application server timeout.

DB2 By default, both DB2 and WebSphere Application Server have the same length timeout, but as the WebSphere Application Server action starts before the DB2 action, it is normally the WebSphere Application Server timeout that is logged.

If one or both of the timeouts have been modified from the default values, and the DB2 timeout is now shorter, the following message is given:

```
AWSJDB803E
An internal deadlock or timeout error has occurred while processing a
database transaction. The internal error message is:
"The current transaction has been rolled back because of a
deadlock or timeout.
Reason code "68".
```

Oracle There is no corresponding timeout on Oracle, so the Dynamic Workload Console hangs.

To resolve the problem, get the database administrator to check if the object in question is locked outside Tivoli Workload Scheduler. If it is, take the appropriate action to unlock it, if necessary asking the database administrator to force unlock the object.

If the object is not locked outside Tivoli Workload Scheduler, retry the operation. If the problem persists contact IBM Software Support for assistance.

On windows, "java process already running" error, restarting the WebSphere Application Server

On windows operating systems, if you try to restart the WebSphere Application Server, after stopWas.cmd, you can have the following error message: java process is already running, even if the stopWas.cmd command returned no error because the WebSphere Application Server java process is still up and running but WASService process is down.

Cause and solution:

The Tivoli Workload Scheduler uses the WASService process to stop and start WebSphere Application Server. If the WASService stops itself, sometimes it is unable to stop java process before goes down and it returns no error, also *appservman* process stops itself and returns no error even if the java process is already up and running.

Resolution: in order to restart the WebSphere Application Server again, you must end the java process associated to WebSphere Application Server from the *Windows Task Manager* panel.

Event management problems

This section describes problems that might occur with processing of events. The topics are as follows:

- “Troubleshooting an event rule that does not trigger the required action”
- “Actions involving the automatic sending of an email fail” on page 131
- “An event is lost” on page 131
- “Expected actions not triggered following an event” on page 131
- “Event rules not deployed after switching event processor” on page 132
- “Event *LogMessageWritten* is not triggered” on page 132
- “Deploy (D) flag not set after ResetPlan command used” on page 133
- “Missing or empty event monitoring configuration file” on page 133
- “Events not processed in correct order” on page 134
- “The *stopeventprocessor* or *switcheventprocessor* commands do not work” on page 134
- “Event rules not deployed with large numbers of rules” on page 134
- “Problem prevention with disk usage, process status, and mailbox usage” on page 135
- File creation and deletion actions not triggered

Troubleshooting an event rule that does not trigger the required action

You have created an event rule but the required action is not triggered when the event condition is encountered.

Cause and solution:

The cause and subsequent solution might be any of a number of things. Use the following check list and procedures to determine what has happened and resolve the problem. The check list uses a test event which has the following characteristics:

```
<eventRule name="TEST1" ruleType="filter" isDraft="no">
  <description>A Rule that checks the sequence of events</description>
  <eventCondition name="fileCreated1" eventProvider="FileMonitor"
    eventType="FileCreated">
    <scope>
      C:\TEMP\FILE5.TXT ON CPU_MASTER
    </scope>
    <filteringPredicate>
      <attributeFilter name="FileName" operator="eq">
        <value>c:\temp\file5.txt</value>
      </attributeFilter>
      <attributeFilter name="Workstation" operator="eq">
        <value>CPU_MASTER</value>
      </attributeFilter>
      <attributeFilter name="SampleInterval" operator="eq">
        <value>60</value>
      </attributeFilter>
    </filteringPredicate>
  </eventCondition>
  <action actionProvider="TWSAction" actionType="sbj" responseType="onDetection">
    <scope>
      SBJ CPU_MASTER#JOB1 INTO CPU_MASTER#JOBS
    </scope>
  </action>
</eventRule>
```

```

<parameter name="JobUseUniqueAlias">
  <value>true</value>
</parameter>
<parameter name="JobDefinitionWorkstationName">
  <value>CPU_MASTER</value>
</parameter>
<parameter name="JobDefinitionName">
  <value>JOB1</value>
</parameter>
</action>
</eventRule>

```

The check list is as follows:

Step 1: Is event management enabled?

Check if the event management feature is enabled (at installation it is enabled by default):

1. Run the following command:

```
optman ls
```

and look for the following entry:

```
enEventDrivenWorkloadAutomation / ed = YES
```

If the value is "YES", go to Step 2: Is the workstation enabled for event processing?.

2. **Action:** If the property is set to *NO*, run the command:

```
optman chg ed=YES
```

3. To effect the change, run:

```
JnextPlan -for 0000
```

Check that the event rule is now being processed correctly. If not, go to Step 2: Is the workstation enabled for event processing?.

Step 2: Is the workstation enabled for event processing?

Check that the workstation is enabled for event processing. By default the master domain manager and backup master domain manager are enabled for event processing, but the default value might have been changed. Do as follows:

1. View the `localopts` file on the master domain manager with a text editor or viewer, and check for the following entry:

```
can be event processor = yes
```

If the value is "yes", go to Step 3: Is the event processor installed, up and running, and correctly configured?.

2. **Action:** If the value is "no", set it to "yes". Save the `localopts` file and stop and start Tivoli Workload Scheduler. Check that the event rule is now being processed correctly. If not, go to Step 3: Is the event processor installed, up and running, and correctly configured?.

Step 3: Is the event processor installed, up and running, and correctly configured?

1. Start **conman**.
2. Issue the **showcpus** command:

```
%sc @!@
```

The output will be similar to the following:

CPUID	RUN	NODE	LIMIT	FENCE	DATE	TIME	STATE	METHOD	DOMAIN
CPU_MASTER	11	*WNT	MASTER	0	0	09/03/07	09:51	I JW MDEA	MASTERDM
FTA1	11	WNT	FTA	0	0		LT		MASTERDM

- Check the STATE field for the presence of an M, a D, and an E (uppercase). In the example, the STATE field has a value of I JW **MDEA**, and the **MDE** is highlighted. If *all* are present, the event processor is installed, up and running, and correctly configured; go to Step 9: Is the SSM agent running (for rules with FileMonitor plug-in-related events only?).
- Actions:** If one or more of M, D, and E are not present, perform one or more of the following actions until they are all present:

The STATE field has neither an uppercase E nor a lowercase e

If there is neither an uppercase E nor a lowercase e, the event processor is not installed. The event processor is installed by default on the master domain manager and backup master domain manager. If you are working on either, then the installation did not complete correctly. Collect the log files in the <TWA_home>/TWS/stdlist directory and contact IBM Software Support for assistance.

The STATE field has a lowercase e

If the STATE field has a lower case e, the event processor is installed but not running. Start the event processor using the **conman startevtproc** command, or the Dynamic Workload Console. If you use **conman**, for example, you will see the following output:

```
%startevtproc
AWSJCL528I The event processor has been started successfully.
```

The STATE field has no M

If the STATE field has no M, **monman** is not running. Start **monman** using the **conman startmon** command. You will see the following output:

```
%startmon
AWSBHU470I A startmon command was issued for CPU_MASTER.
```

The STATE field has no D

If the STATE field has no D, the current monitoring package configuration is not deployed. Go to Step 5: Has the rule been added to the monitoring configuration on the workstation?.

- Rerun the **showcpus** command.
- When the M, D, and E are all present, check that the event rule is now being processed correctly. If not, go to Step 9: Is the SSM agent running (for rules with FileMonitor plug-in-related events only?).

Step 4: Is the workstation definition present in the plan?

- Start **conman**.
- Issue the **showcpus** command:

```
%sc @!@
```

If the workstation definition is not included in the plan, add it. For more information about how to add a workstation to the plan, see the *Administration Guide*. If the workstation definition is included, go to Step 5: Has the rule been added to the monitoring configuration on the workstation?.

Step 5: Has the rule been added to the monitoring configuration on the workstation?

1. Check if the rule is present in the workstation monitoring configuration by running the **conman showcpus** command with the **;getmon** argument:

```
%sc ;getmon
Monitoring configuration for CPU_MASTER:
*****
*** Package date : 2008/09/03 07:48 GMT ***
*****

TEST1::FileMonitor#FileCreated:C:\TEMP\FILE5.TXT ON CPU_MASTER;
TEST1::TWSObjectsMonitor#JobSubmit:* # * . TEST*;
```

If the rule is present, go to Step 7: Has the new monitoring configuration been deployed to the workstation?.

2. **Action:** If the configuration does not contain the expected rule, go to Step 6: Is the rule active.

Step 6: Is the rule active

If the configuration does not contain the expected rule, check if it is active.

1. Check the rule status, using the **composer list** command or the Dynamic Workload Console. If you use composer, for example, you will see output similar to the following:

```
-list er=@
```

Event Rule Name	Type	Draft	Status	Updated On	Locked By
TEST1	filter	N	active	09/03/2008	-

If the rule is in *active* status go to Step 7: Has the new monitoring configuration been deployed to the workstation?.

2. **Action:** If the rule is in *error* status, activate the Tivoli Workload Scheduler trace, collect the log files in the *<TWA_home>/TWS/stdlist* directory and contact IBM Software Support for assistance.

Step 7: Has the new monitoring configuration been deployed to the workstation?

If the rule is active, check if the new monitoring configuration has been deployed to the workstation.

1. The deployment of a new monitoring configuration can be checked in either of these ways:
 - Check in the *<TWA_home>/TWS/monconf* if the configuration is present
 - Check in the SystemOut file in *<TWA_home>/WAS/TWSPprofile/logs/server1*. Look for the message:

```
[9/3/07 9:50:00:796 CEST] 00000020 sendEventReadyConfiguration(wsInPlanIds, zipsToDeploy)
AWS000001I The workstation "CPU_MASTER" has been notified about
a new available configuration.
```

If the message is present for the workstation in question after the time when the rule was made available for deployment, then the new configuration has been deployed.

If the configuration has been deployed, go to Step 8: Has the deploy of the new monitoring configuration worked correctly?.

2. **Action:** If the configuration has not been deployed, deploy it with the **conman deploy** command:


```
%deploy
AWSBHU470I A deployconf command was issued for MASTER_CPU.
```

Check that the event rule is now being processed correctly. If not, go to Step 8: Has the deploy of the new monitoring configuration worked correctly?.

Step 8: Has the deploy of the new monitoring configuration worked correctly?

If the new monitoring configuration has been deployed, check that the deployment was successful:

1. Check in the `<TWA_home>/TWS/stdlist/traces/<date>_TWSMERGE.log`, and look for the most recent occurrence of these 2 messages:

```
09:51:57 03.09.2008|MONMAN:INFO:=== DEPLOY ===> CPU_MASTER has been notified
of the availability of the new monitoring configuration.
09:51:57 03.09.2008|MONMAN:INFO:=== DEPLOY ===> The zip file d:\TWS\twsuser\monconf\deployconf.zip
has been successfully downloaded.
```

If you find these messages, referring to the workstation in question, and occurring after the time when the rule was deployed, then the rule has been successfully deployed to the workstation: go to Step 9: Is the SSM agent running (for rules with FileMonitor plug-in-related events only?).

2. **Actions:** If you find messages that indicate an error, follow one of these actions:

Message indicates that the server could not be contacted or that the action has been resubmitted by monman

The message you find is either of the following:

```
=== DEPLOY ===> ERROR contacting the server for receiving the zip file (rc=8)
=== DEPLOY ===> The deploy action has been automatically resubmitted by monman.
```

The application server could be down. Either wait for 5 minutes, or follow the instructions about how to use **appserverman** (see *Tivoli Workload Scheduler: Administration Guide*) to determine if the application server is down, and if it is being restarted automatically, or needs to be restarted manually.

If you need to change any aspect of the application server configuration, run **JnextPlan -for 0000**.

When you are certain that the application server is up, retry Step 8: Has the deploy of the new monitoring configuration worked correctly?.

Message indicates a problem with decoding or unzipping the zip

The message you find is either of the following:

```
=== DEPLOY ===> ERROR decoding the zip file temporarily downloaded in
                    <TWA_home>/TWS/monconf
=== DEPLOY ===> ERROR unzipping the zip file <file_name>
```

Collect the log files and contact IBM Software Support for assistance.

Step 9: Is the SSM agent running (for rules with FileMonitor plug-in-related events only?)

1. If the rule has an event that uses the FileMonitor plug-in, check that the SSM Agent is running. Check in the log that when the **conman startmon** command was run (either when you ran it manually or when Tivoli Workload Scheduler started).

2. Then look ahead in the log for the following message:

```
11:13:56 03.09.2008|MONMAN:INFO:SSM Agent service successfully started
```

If it is present, or the rule does not use the FileMonitor plug-in, go to Step 6: Is the rule active.

3. **Action:** If the SSM Agent message is not present, collect the log files in the <TWA_home>/TWS/stdlist directory and the <TWA_home>/TWS/ssm/ directory and contact IBM Software Support for assistance.

Step 10: Have the events been received?

You know the rule has been deployed, but now you need to know if the event or events have been received.

1. Check in the SystemOut of the server to see if the event has been received. The output is different, depending on the type of event:

FileMonitorPlugIn event

- a. This is the output of a FileMonitorPlugIn event:

```
[9/3/07 9:55:05:078 CEST] 00000035 EventProcessor A com.ibm.tws.event.EventProcessorManager
processEvent(IEvent)
AWSEVP001I The following event has been received:
event type = "FILECREATED"; event provider = "FileMonitor";
event scope = "c:\temp\file5.txt on CPU_MASTER".
FILECREATED FileMonitor c:\temp\file5.txt on CPU_MASTER
```

If the event has been received, go to Step 11: Has the rule been performed?.

- b. If the event has not been received check if it has been created by looking in the traps.log for the message that indicates that the event has been created:

```
.1.3.6.1.4.1.1977.47.1.1.4.25 OCTET STRING FileCreatedEvent event
```

- c. **Action:** Whether the event has or has not been created, collect the information in the <TWA_home>/TWS/ssm directory and contact IBM Software Support for assistance.

TWSObjectMonitorPlugIn event

- a. This is the output of a TWSObjectMonitorPlugIn event:

```
[9/3/07 12:28:38:843 CEST] 00000042 EventProcesso A com.ibm.tws.event.EventProcessorManager
processEvent(IEvent)
AWSEVP001I The following event has been received: event type = "JOBSUBMIT";
event provider = "TWSObjectsMonitor"; event scope = "CPU_MASTER # JOBS .
(CPU_MASTER #) TEST". JOBSUBMIT "TWSObjectsMonitor" CPU_MASTER # JOBS .
(CPU_MASTER #) TEST
```

- b. **Action:** If the event has not been received, collect the log data and contact IBM Software Support for assistance.
- c. If the TWSObjectMonitorPlugIn event has been received, check in the same log that the EIF event has been sent. This is the output of an EIF event:

```
12:27:18 03.09.2008|MONMAN:INFO:Sending EIF Event:
"JobSubmit;
TimeStamp="2008-09-03T12:26:00Z/";
EventProvider="TWSObjectsMonitor";
HostName="CPU_MASTER";
IPAddress="9.71.147.38";
```

```

PlanNumber="11";
Workstation="CPU_MASTER";
JobStreamWorkstation="CPU_MASTER";
JobStreamId="JOBS";
JobStreamName="JOBS";
JobStreamSchedTime="2008-09-03T12:26:00";
JobName="TEST";
Priority="10";
Monitored="false";
EstimatedDuration="0";
ActualDuration="0";
Status="Waiting";
InternalStatus="ADD";
Login="twuser";END

```

- d. If the EIF event has been sent, it might be cached in the `<TWA_home>/TWS/EIF` directory.
- e. If the event is found there, check the communication with the agent and the server. If no communication problem is present wait until the event is sent.
- f. The event might also be cached in the machine where the event processor is located. Check this in the path `<WAS_profile_path>/temp/TWS/EIFListener`, where the default path for `<WAS_profile_path>` is `TWA_home/WAS/TWSprofile`. If the event is found there, check the communication with the agent and the server. If no communication problem is present wait until the event is sent.

2. **Action:** If the problem persists, collect the log data and contact IBM Software Support for assistance.

Step 11: Has the rule been performed?

You now know that the event has been received, but that the action has apparently not been performed.

1. Check in the SystemOut of the server to see if the rules have been performed. Look for messages like these:

```

[9/3/07 9:55:05:578 CEST] 00000035 ActionHelper A com.ibm.tws.event.plugin.action.ActionHelper
invokeAction(ActionContext,Map,EventRuleHeader)
AWSAHL004I The rule "TEST1" has been triggered. TEST1
[9/3/07 9:55:05:625 CEST] 00000036 ActionHelper A com.ibm.tws.event.plugin.action.ActionHelper
AsynchAction::run()
AWSAHL002I The action "sbj" for the plug-in "TWSAction" has been started.
sbj TWSAction
[9/3/07 9:55:06:296 CEST] 00000036 ActionHelper A com.ibm.tws.event.plugin.action.ActionHelper
AsynchAction::run()
AWSAHL003I The action "sbj" for the plug-in "TWSAction" has completed.
sbj TWSAction

```

If the rule has been triggered and the action completed, go to Step 12: Is the problem in the visualization of the event?.

2. **Action:** If the action has not been completed collect the log data and contact IBM Software Support for assistance.

Step 12: Is the problem in the visualization of the event?

Action: If the event has been received, but you cannot see it, there might be a problem with the console you are using to view the event. See Chapter 10, "Troubleshooting Dynamic Workload Console problems," on page 155.

Actions involving the automatic sending of an email fail

An event rule is created, including as the required action the sending of an email. When the event occurs, the action fails with the following message:

```
AWSMSP104E The mail "<mailID>" has not been successfully
           delivered to "<recipient>".
Reason: "Sending failed;
nested exception is:
?????class javax.mail.MessagingException: 553 5.5.4 <TWS>...
Domain name required for sender address TWS
```

Cause and solution:

The mail send action failed because the domain name of the SMTP server was not defined in the mail sender name global option: mailSenderName (ms).

Use the **optman** command to specify the correct mail sender name including the domain. For example, if the mail sender name is tws@alpha.ibm.com, issue the following command:

```
optman chg ms tws@alpha.ibm.com
```

An event is lost

You have sent a large number of events to the event processor. When you check the event queue you find that the most recent event or events are missing.

Cause and solution:

The event queue is not big enough. The event queue is circular, with events being added at the end and removed from the beginning. However, if there is no room to write an event at the end of the queue it is written at the beginning, overwriting the event at the beginning of the queue.

You cannot recover the event that has been overwritten, but you can increase the size of the queue to ensure the problem does not recur. Follow the instructions in "Managing the event queue" in *Tivoli Workload Scheduler: Administration Guide*.

Expected actions not triggered following an event

In situations where a large number of events are generated and where an action or actions that are expected to be triggered on the master are not triggered, then most probably not all events arrived on the server. There are a number of steps you can perform to collect the necessary information IBM Software Support requires to assist you in solving the problem.

Collecting required information:

1. Set the keyword **LogEvents** = YES in the following files on the server and client. By default, the value assigned to this keyword is NO.

On the client

- *TWA_HOME*/ssm/eif/tecad_snmp_eeif.tmp1
- *TWSA_HOME*/EIF/monmaneif.tmp1

On the server

```
<WAS_profile_path>/temp/TWS/EIFListener/eif.tmp1
```

where the default path for <WAS_profile_path> is *TWA_home*/WAS/TWSprofile.

2. On the client, stop and start monman by submitting the respective command.

Stop **conman stopmon**

Start **conman startmon**

3. On the server, stop and start the event processor submitting the respective command.

Stop **conman stoeventproc**

Start **conman startevertproc**

4. All generated events are logged in a file. By default, this file is stored in the path *TWA_home/EIF*. Collect these logs and send them to IBM support. Because this log file can become quite large very quickly, there are ways to filter the events that get logged in the file.

On the server, set the **LogEventsFilter** keyword using a regular expression to filter events written on the server. By default, the value of this keyword is `".*"` and therefore all events are logged. For example, if you are aware that problems originate from a specific client, then you can specify the CPU name of the client in the value of the **LogEventsFilter** keyword to limit the events logged to the file to this specific client as follows:

`LogEventsFilter=.*<cpu_name>#MONMAN.*`

Event rules not deployed after switching event processor

You have switched the event processor, but new or amended rules have not been deployed (the event states of the workstations that were affected by the new or amended rules do not show "D" indicating that the rules are not up-to-date, and the **getmon** command shows the old rules).

Cause and solution:

The probable cause is that you made some changes to the rules before running the **switcheventprocessor** command, and these rules were not deployed (for whatever reason) before the switch.

To remediate the situation, run the command **conman deployconf** `<workstation_name>`, for each affected workstation, and the rule changes will be deployed.

To avoid that this problem reoccurs, run **planman** with the *deploy* action before running **switcheventprocessor**.

Event *LogMessageWritten* is not triggered

You are monitoring a log file for a specific log message, using the *LogMessageWritten* event. The message is written to the file but the event is not triggered.

Cause and solution:

The SSM agent monitors the log file. It sends an event when a new message is written to the log file that matches the string in the event rule. However, there is a limitation. It cannot detect the very latest message to be written to the file, but only messages prior to the latest. Thus, when message line "n" is written containing the string that the event rule is configured to search for, the agent does not detect that a message has been written, because the message is the last one in

the file. When any other message line is written, if or not it contains the monitored string, the agent is now able to read the message line containing the string it is monitoring, and sends an event for it.

There is no workaround to resolve this problem. However, it should be noted that in a typical log file, messages are being written by one or other processes frequently, perhaps every few seconds, and the writing of a subsequent message line will trigger the event in question. If you have log files where few messages are written, you might want to attempt to write a dummy blank message after every "real" message, in order to ensure that the "real" message is never the last in the file for any length of time.

Deploy (D) flag not set after ResetPlan command used

The deploy (D) flag is not set on workstations after the **ResetPlan** command is used.

Cause and solution:

This is not a problem that affects the processing of events but just the visualization of the flag which indicates that the event configuration file has been received at the workstation.

No action is required, because the situation will be normalized the next time that the event processor sends an event configuration file to the workstation.

However, if you want to take a positive action to resolve the problem, do the following:

- Create a dummy event rule that applies only to the affected workstations
- Perform a **planman deploy** to send the configuration file
- Monitor the receipt of the file on the agent
- When it is received, delete the dummy rule at the event processor

Missing or empty event monitoring configuration file

You have received a MONMAN trace message on a workstation, similar to this:

```
MONMAN:INFO:=== DEPLOY ==> ERROR reading the zip file
                  /home/f_edwa3/monconf/deployconf.zip.
                  It is empty or does not exist".
```

Cause and solution:

The Tivoli Workload Scheduler agent on a workstation monitors for events using a configuration file. This file is created on the event processor, compressed, and sent to the agent. If a **switcheventprocessor** action is performed between the creation of the file on the old event processor and the receipt on the new event processor of the request for download from the agent, the file is not found on the new event processor, and this message is issued.

To resolve the problem, do the following:

- Create a dummy event rule that applies only to the affected workstation
- Perform a **planman deploy** to send the configuration file
- Monitor the receipt of the file on the agent
- When it is received, delete the dummy rule at the event processor

Events not processed in correct order

You have specified an event rule with two or more events that must arrive in the correct order, using the *sequence* event grouping attribute. However, although the events occurred in the required sequence the rule is not triggered, because the events arrived at the event processor in an order different from their creation order.

Cause and solution:

Events are processed in the order they arrive, not the order they are created. If they arrive in order different from the creation order, you will not get the expected result.

For example, consider a rule which is triggered if event A defined on workstation AA occurs before event B which is defined on workstation BB. If workstation AA loses its network connection before event A occurs, and does not regain it until after event B has arrived at the event processor, the event rule will not be satisfied, even though the events might have occurred in the correct order.

The solution to this problem is that if you need to define a rule involving more than one event, use the *set* event grouping attribute, unless you can be certain that the events will arrive at the event processor in the order they occur.

The `stopeventprocessor` or `switcheventprocessor` commands do not work

You have run `stopeventprocessor` or `switcheventprocessor` but the command has failed. The log indicates a communication problem.

Cause and solution:

If you issue the `stopeventprocessor` command from a workstation other than that where the event processor is configured, the command uses the command-line client, so the user credentials for the command-line client must be set correctly.

Similarly, if you use `switcheventprocessor`, it also uses the command-line client, so the user credentials for the command-line client must be set correctly also in this case.

Event rules not deployed with large numbers of rules

You have run `planman deploy` (or the equivalent action from the Dynamic Workload Console), with a very large number of event rules, but the command has failed. The log indicates a memory error.

Cause and solution:

A large number of event rules requires a Java heap size for the application server larger than the default. In this context, a large number would be 10 000 or more. Doubling the default size should be sufficient.

Full details of how to do this are described in the *Tivoli Workload Scheduler: Administration Guide* in the section on *Increase application server heap size* in the *Performance* chapter.

Problem prevention with disk usage, process status, and mailbox usage

You can use event-driven workload automation (EDWA) to monitor the health of the Tivoli Workload Scheduler environment and to start a predefined set of actions when one or more specific events take place. You can prevent problems in the Tivoli Workload Scheduler environment by monitoring the filling percentage of the mailboxes, the status of Tivoli Workload Scheduler processes, and the disk usage of the Tivoli Workload Scheduler file system.

Full details of how to do this are described in the *Tivoli Workload Scheduler: Administration Guide*, as follows:

- section on *Monitoring the disk space used by Tivoli Workload Scheduler* in the *Data maintenance* chapter
- sections on *Monitoring the size of Tivoli Workload Scheduler message queues* and *Monitoring the status of Tivoli Workload Scheduler processes* in chapter *Network administration*

See also “Configuring trace properties when the agent is running” on page 34.

Problems using the "legacy" global options

This section describes problems that might occur when running Tivoli Workload Scheduler with the "legacy" global options set. The "legacy" global options are those that have the word "Legacy" in their option name in **optman**. Use them if you want to maintain certain Tivoli Workload Scheduler behaviors as they were in previous versions of Tivoli Workload Scheduler.

- “Time zones do not resolve correctly with enLegacyStartOfDayEvaluation set”
- “Dependencies not processed correctly when enLegacyId set”

Time zones do not resolve correctly with enLegacyStartOfDayEvaluation set

You are using Tivoli Workload Scheduler with the enLegacyStartOfDayEvaluation and enTimeZone options set to *yes* to convert the startOfDay time set on the master domain manager to the local time zone set on each workstation across the network. You submit a job or job stream with the **at** keyword, but the job or job stream does not start when expected.

Cause and solution:

Add the **absolute** keyword to make sure that the submission times are resolved correctly. The **absolute** keyword specifies that the start date is based on the calendar day rather than on the production day.

Dependencies not processed correctly when enLegacyId set

You are using Tivoli Workload Scheduler in a network which includes agents running on versions older than 8.3, but managed by a version 8.3 or later master domain manager, with the enLegacyId option set to *yes*, to enable the use of the former job stream ID format. When you create multiple instances of a job stream as pending predecessors, errors caused by identification problems at submission time are given.

Cause and solution:

There is no workaround to this other than to upgrade the agents to the level of the master domain manager.

Managing concurrent accesses to the Symphony file

This section contains two sample scenarios describing how Tivoli Workload Scheduler manages possible concurrent accesses to the Symphony file when running **stageman**.

Scenario 1: Access to Symphony file locked by other Tivoli Workload Scheduler processes

If Tivoli Workload Scheduler processes are still active and accessing the Symphony file when **stageman** is run, the following message is displayed:

```
Unable to get exclusive access to Symphony.  
Shutdown batchman and mailman.
```

To continue, stop Tivoli Workload Scheduler and rerun **stageman**. If **stageman** aborts for any reason, you must rerun both **planman** and **stageman**.

Scenario 2: Access to Symphony file locked by stageman

If you try to access the plan using the command-line interface while the Symphony is being switched, you get the following message:

```
Current Symphony file is old. Switching to new Symphony.  
Schedule mm/dd/yyyy (nnnn) on cpu, Symphony switched.
```

StartApp Server problems

The **StartApp Server** command checks if the WebSphere Application Server is running, if the WebSphere Application Server is not running **startApp Server** starts it.

In case of failure:

- Rerun the job.

MakePlan problems

MakePlan performs the following actions:

- Replans or extends the preproduction plan.
- Produces the Symnew file.
- Generates preproduction reports in the joblog

The following problems could be encountered when running **MakePlan**:

- "MakePlan fails to start" on page 137
- "Unable to establish communication with the server on host - AWSBEH023E" on page 137
- "The user "twuser" is not authorized to access the server on host - AWSBEH021E" on page 137
- "The database is already locked - AWSJPL018E" on page 137
- "An internal error has occurred - AWSJPL006E" on page 137
- "The production plan cannot be created - AWSJPL017E" on page 138
- "An internal error has occurred - AWSJPL704E" on page 138

MakePlan fails to start

If MakePlan fails to start:

- Global lock might be left to 'set'. Use **planman unlock** to 'reset' it
- Rerun the job to recover:
 - Preproduction plan is automatically reverified and updated.
 - Symnew is recreated.

How to stop it:

- Stopping the job might not stop the processing that is still running on WebSphere Application Server or in the database.
- Force the Database statement to close if a database statement runs for too long and causes **MakePlan** to abend.
- Restart the WebSphere Application Server if processing is still running on WebSphere Application Server and **MakePlan** does not finish.

Note: Check if the database statistics are enabled. If not, it is recommended to schedule the runstatistics script stored in the dbtools Tivoli Workload Scheduler directory.

Unable to establish communication with the server on host - AWSBEH023E

If you receive the following error message from **MakePlan** stdlist:

```
AWSBEH023E Unable to establish communication with the server on host
"127.0.0.1" using port "31116"
```

Cause and solution: This error means that the application server is down and **MakePlan** cannot continue. If this happens, start the WebSphere Application Server and check the WebSphere Application Server logs to identify the reason why the WebSphere Application Server has stopped.

The user "twuser" is not authorized to access the server on host - AWSBEH021E

```
AWSBEH021E The user "twuser" is not authorized to access the server on
host "127.0.0.1" using port "31116"
```

Cause and solution: This is an authorization error. Check your Tivoli Workload Scheduler user name and password in the usersopts file.

The database is already locked - AWSJPL018E

```
AWSJPL018E The database is already locked
```

Cause and solution: The previous operation of **MakePlan** is stopped and the global lock is not reset. To recover the situation run **planman unlock**.

An internal error has occurred - AWSJPL006E

```
AWSJPL006E An internal error has occurred. A database object "xxxx"
cannot be loaded from the database.
```

Cause and solution: Usually "xxxx" is an object like a workstation, job, or job

stream. This error means that a connection with the database is broken. In this case, check the error in the SystemOut.log and the ffdc directory as additional information related to the error is logged there.

The production plan cannot be created - AWSJPL017E

AWSJPL017E The production plan cannot be created because a previous action on the production plan did not complete successfully. Refer to the message help for more details.

Cause and solution: This error might mean that a previous operation on the preproduction plan is performed but ended with an error. Generally it is present when "ResetPlan - scratch" is performed but does not end successfully.

An internal error has occurred - AWSJPL704E

AWSJPL704E An internal error has occurred. The planner is unable to extend the preproduction plan.

Cause and solution: This error might mean that **MakePlan** cannot extend the preproduction plan. Different root causes are associated with this issue, typically always related to the database, for example, no space for the tablespace or full transaction logs. Check for more information in the SystemOut.log directory or in the directories where ffdc (*first failure data capture*) utility saves the extracted information.

SwitchPlan problems

SwitchPlan performs the following actions:

- Stops all the workstations
- Runs **Stageman** to:
 - Merge the old Symphony file with SymNew
 - Archive the old Symphony file in the schedlog directory
- Runs the **planman confirm** command to update the database plan status information . For example, the plan end date and the current run number.
- Restarts the master to distribute the Symphony file and restart scheduling.

The following problems could be encountered when running **SwitchPlan**:

- "When SwitchPlan fails to start"
- "The previous Symphony file and Symnew file have the same run number - AWSBHV082E" on page 139

When SwitchPlan fails to start

If **SwitchPlan** fails to start:

1. **planman confirm** is not running. Perform the following actions:
 - a. Check the logs
 - b. Run "**planman showinfo**"
 - c. Rerun **SwitchPlan**
2. **planman confirm** failed. Perform the following action:
 - Manually run **planman confirm** and **conman confirm**.
3. **planman confirm** was already run and the plan end date has been updated. Perform the following action:
 - Run **conman start**

If **conman stop** hangs, kill the **conman** command. This might impact the plan distribution because it stops the agents left running before distributing the new Symphony.

The previous Symphony file and Symnew file have the same run number - AWSBHV082E

If **SwitchPlan** stdlist shows the following messages:

```
STAGEMAN: AWSBHV082E: The previous Symphony file and Symnew file have
the same run number. They cannot be merged to form the new symphony file.
```

Cause and solution: There are several possible causes because the Symphony and Symnew run numbers have the same values, common causes for this are:

1. **MakePlan** did not extend the run number in the Symnew file.
2. **SwitchPlan** ran before **MakePlan**.
3. The **Stageman** process ran twice on the same Symnew file without resetting the plan or deleting the Symphony file.

```
AWSJCL054E: The command "CONFIRM" has failed.
```

```
AWSJPL016E: An internal error has occurred. A global option "confirm
run member" cannot be set.
```

Cause and solution: These error messages are present when the last step of the **SwitchPlan**, that is **planman confirm** fails. Analyze the SystemOut.log for more information and rerun **planman confirm**.

Create Post Reports

Create Post Reports has the following function:

- General post production reports in the job output.

In case of failure:

- Rerun the job if reports are needed.

Update Stats problems

Update Stats has the following functions:

- Runs logman to update job statistics and history.
- Extends the Preproduction plan if it's length is shorter than minLen.

In case of failure:

- Rerun the job or manually run "**logman <file>** " on the latest schedlog file.
- If it does not run, the statistics and history will be partial. Pre production plan is updated at the beginning of **MakePlan** .

How to stop it:

- Kill the job or logman process, the statistics and history will be partial until the job or logman is rerun.

Miscellaneous problems

The following problems might occur:

- “An error message indicates that a database table, or an object in a table, is locked”
- “Command line programs (like composer) give the error "user is not authorized to access server"" on page 141
- “The rmstdlist command gives different results on different platforms” on page 141
- “Question marks are found in the stdlist” on page 141
- “Deleting stdlist or one of its files when processes are still running” on page 142
- “A job with a "rerun" recovery job remains in the "running" state” on page 142
- “Job statistics are not updated daily” on page 143
- “A Job Scheduler dependency is not added” on page 143
- “Incorrect time-related status displayed when time zone not enabled” on page 143
- “Completed job or job stream not found” on page 143
- “Variables not resolved after upgrade” on page 144
- “Default variable table not accessible after upgrade” on page 144
- “Local parameters not being resolved correctly” on page 144
- “Log files grow abnormally large in mixed environment with version 8.4 or higher master domain manager and 8.3 or lower agents” on page 145
- “Deleting leftover files after uninstallation is too slow” on page 146
- “Corrupted special characters in the job log from scripts running on Windows” on page 147
- “Failover Cluster Command Interface deprecated” on page 147

An error message indicates that a database table, or an object in a table, is locked

An error message indicates that a function cannot be performed because a table, or an object in a table, is locked. However, the table or object does not appear to be locked by another Tivoli Workload Scheduler process.

Cause and solution:

The probable cause is that a user has locked the table by using the database command-line or GUI:

DB2 Just opening the DB2 GUI is sufficient to lock the database tables, denying access to all Tivoli Workload Scheduler processes.

Oracle If the Oracle command-line is opened without the auto-commit option, or the GUI is opened, Oracle locks all tables, denying access to all Tivoli Workload Scheduler processes.

To unlock the table close the command-line or GUI, as appropriate.

Note: Tivoli Workload Scheduler provides all of the database views and reports you need to manage the product. You are strongly recommended to not use the facilities of the database to perform any operations, including viewing, on the database tables.

Command line programs (like composer) give the error "user is not authorized to access server"

You launch CLI programs (like composer) but when you try and run a command, the following error is given:

```
user is not authorized to access server
```

Cause and solution:

This problem occurs when the user running the command has a null password. Composer, and many of the other Tivoli Workload Scheduler CLI programs cannot run if the password is null.

Change the password of the user and retry the operation.

The rmstdlist command gives different results on different platforms

The rmstdlist command on a given UNIX platform gives results that differ from when it is used on other platforms with the same parameters and scenario.

Cause and solution:

This is because on UNIX platforms the command uses the *-mtime* option of the **find** command, which is interpreted differently on different UNIX platforms.

To help you determine how the *-mtime* option of the **find** command is interpreted on your workstation, consider that the following command:

```
<TWA_home>/TWS/bin/stdlist/rmstdlist -p 6
```

gives the same results as these commands:

```
find <TWA_home>/TWS/stdlist/ -type d ! -name logs ! -name traces -mtime +6 -print
```

```
find <TWA_home>/TWS/stdlist/logs/ -type f -mtime +6 -print
```

```
find <TWA_home>/TWS/stdlist/traces/ -type f -mtime +6 -print
```

Look at your operating system documentation and determine how the option works.

The rmstdlist command fails on AIX with an exit code of 126

The rmstdlist command on AIX fails with an exit code of 126 and no other error message.

Cause and solution:

This could be because there are too many log files in the stdlist directory.

On AIX, you should regularly remove standard list files every 10-20 days. See the usage instructions in the *Tivoli Workload Scheduler: User's Guide and Reference* for full details.

Question marks are found in the stdlist

You discover messages in the log or trace files that contain question marks. For example the following (the message has been split over several lines to make it more readable - the question marks are highlighted to make them more obvious):

```
10:20:02 03.02.2008|BATCHMAN:+ AWSBHT057W
Batchman has found a non-valid run number in the Symphony
file for the following record type: "Jt" and object:
"F235011S3_01#???[()],(0AAAAAAAAAAAAAZD)].A_7_13 (#J18214)".
```

Cause and solution:

This problem occurs when the process that needs to write the log message cannot obtain the Job Scheduler name. For example, when a Job Scheduler is dependent on a Job Scheduler that is not in the current plan (Symphony file). The process writes "???" in place of the missing Job Scheduler name.

The message contains the Job Scheduler ID (in the above example it is the string in the second set of parentheses: (0AAAAAAAAAAAAAZD)). Use the Job Scheduler ID to identify the instance of the Job Scheduler, and take any action suggested by the message that contained the question marks.

Deleting stdlist or one of its files when processes are still running

```
|      Erroneously deleted the stdlist directory or one of its files while processes are
|      running.
```

```
|      You erroneously deleted the stdlist directory or one of its files while processes
|      are running and you receive the following error when performing an operation:
|      Permission denied
|      Bad file descriptor
```

Cause and solution:

This problem occurs because the directories or files with root ownership are not re-created during the initialization phase.

```
|      According to your operating system, perform one of the following actions:
```

UNIX

- Create the directory or file that you deleted with **twuser** and **group** ownership.
- Modify the ownership of the directory or file created with **root** ownership to **twuser** and **group** ownership.

Windows

- Ensure that you are running the command line with the **Run as Administrator** privilege level.

A job with a "rerun" recovery job remains in the "running" state

You have run a job specifying a recovery job using the "rerun" recovery method. The original job fails, but when the recovery job starts the original job shows that the recovery action has been completed successfully, but remains in the "running" state.

Cause and solution:

This problem would occur if the recovery job was specified to run on a different workstation and domain from the original job. The original job is then unable to detect the state of the recovery job, so it cannot determine if the recovery job has finished or what state it finished in.

To resolve the problem for the specific job that is still in "running" state, you must manually stop the job.

To avoid the recurrence of the problem specify the "rerun" recovery action on the same workstation in the same domain.

Job statistics are not updated daily

Job statistics are not updated daily, as they were with versions prior to version 8.3.

Cause and solution:

Job statistics are updated by **JnextPlan**. If you are running **JnextPlan** less frequently than daily, the statistics are only updated when **JnextPlan** is run.

A Job Scheduler dependency is not added

A dependency is added to a Job Scheduler instance and the Job Scheduler is saved. When the list of dependencies is reopened, the new dependency is not present.

Cause and solution:

This occurs when a Job Scheduler instance already has the maximum number (40) of dependencies defined. Normally, an error message would alert you to the limit, but the message might not be displayed if there is a delay propagating the Symphony updates across the network or if your update coincided with updates by other users.

Incorrect time-related status displayed when time zone not enabled

You are using Tivoli Workload Scheduler in an environment where nodes are in different time zones, but the time zone feature is not enabled. The time-related status of a job (for example, "Late") is not reported correctly on workstations other than that where the job is being run.

Cause and solution:

Enable the time zone feature to resolve this problem. See *Tivoli Workload Scheduler: User's Guide and Reference* to learn more about the time zone feature. See *Tivoli Workload Scheduler: Administration Guide* for instructions on how to enable it in the global options.

Completed job or job stream not found

A job or job stream that uses an alias has completed but when you define a query or report to include it, the job or job stream is not included.

Cause and solution:

Jobs and job streams in *final* status are stored in the archive with their *original* names, not their aliases, so any search or reporting of completed jobs must ignore the aliases.

Variables not resolved after upgrade

After upgrading to version 8.5, global variables are not resolved.

Cause and solution:

During the upgrade to version 8.5, all the security file statements relating to your global variables were copied by the install wizard into a default variable table in the new security file. Global variables are disabled in version 8.5, and can only be used through the variable tables. If you subsequently rebuilt the security file using the output from your *previous* **dumpsec** as input to the new **makesec**, you will have overwritten the security statements relating to your default variable table, so no user has access to the default variable table.

If you have a backup of your security file from prior to when you ran **makesec**, run **dumpsec** from that, and merge your old **dumpsec** output file with your new one, as described in the upgrade procedure in the *Tivoli Workload Scheduler: Planning and Installation Guide*.

If you do not have a backup, create the default variable table security statement, following the instructions about configuring the security file in the *Tivoli Workload Scheduler: Administration Guide*.

Default variable table not accessible after upgrade

After upgrading to version 8.5, your default variable table is not accessible by any user.

Cause and solution:

This problem has exactly the same Cause and solution: as the preceding - see "Variables not resolved after upgrade."

Local parameters not being resolved correctly

You have scheduled a job or job stream that uses local parameters, but the parameters are not resolved correctly.

Cause and solution:

One reason for this could be that one or both of the files where the parameters are stored have been deleted or renamed.

Check that the following files can be found in the *TWA_home*/TWS directory:

parameters
parameters.KEY

These files are required by Tivoli Workload Scheduler to resolve local parameters, so they must not be deleted or renamed. Fix the problem as follows:

1. If the files have been renamed, rename them to the original names.
2. If the files have been deleted, recreate them, using the **parms** utility.
3. To make the changes effective, restart the application server, using the **stopappserver** and **startappserver** commands.

Log files grow abnormally large in mixed environment with version 8.4 or higher master domain manager and 8.3 or lower agents

The problem occurs in mixed environments where Tivoli Workload Scheduler agents version 8.3 or earlier run under a master domain manager version 8.4 or later. The problem is that the older version agents do not correctly handle the Tivoli Workload Scheduler events generated by the features added by version 8.4 and later, such as Event Driven Workload Automation (monman), Workload Service Assurance (critical path), and WebSphere Application Server manager (appservman). This may cause random execution, duplication of Tivoli Workload Scheduler events or dumping of Tivoli Workload Scheduler event records type "00" that flood the log files.

The cure to this problem is to install on your older version agents the corresponding fix pack containing the fix for APAR IZ62730.

An alternative to installing the fix pack on your agents is to apply the following workaround on your version 8.4 or later master domain manager, provided your master runs one of the following product versions:

- 8.4 with fix pack 5 or later
- 8.5 with fix pack 1 or later
- 8.5.1 with fix pack 1 or later

Follow these steps:

1. Disable the Event Driven Workload Automation (EDWA) feature
 - `optman chg ed=no`
2. Check that EDWA is actually disabled
 - `optman ls`
 - >>>> `enEventDrivenWorkloadAutomation / ed = NO`
3. Shut down Tivoli Workload Scheduler and WebSphere Application Server
4. Delete the Mailbox.msg file because it contains messages related to stopping the appservman process
5. Enable new behavior of appservman by adding to the localopts file the following key:
 - Appserver disable send event = yes
6. Start up Tivoli Workload Scheduler
7. Check that the broadcast of newer product versions (8.4 and later) events is actually disabled by looking for the following message in the `<TWS_home>/stdlist/traces/TWSMERGE.log`: *"Broadcasting of Appservman events is disabled"*

If you cannot find this message, the reason is that your master is not patched with the fix pack version listed above. If this is the case, you can run the following recovery procedure (but this will preclude appservman from starting):

1. Shut down Tivoli Workload Scheduler and WebSphere Application Server
2. Delete the Mailbox.msg file because it contains messages related to the start up of appservman
3. Start up WebSphere Application Server without the appservman process:
 - `<TWSHOME>/wastools/StartWas.sh -direct`
4. Start up Tivoli Workload Scheduler without the *appservman* process
 - `Startup -noappsrv`

The master domain manager is now ready to create a plan without the Event Driven Workload Automation. You can wait for the next JnextPlan or run:
JnextPlan -for 000

If you have a mix of version 8.3 and version 8.4 agents, follow these steps:

1. Unlink and shut down only the version 8.4 agents
2. Check that no Tivoli Workload Scheduler processes are running

```
ps -fu <TWS_user>
```
3. Delete the Mailbox.msg file because it contains messages related to the monman process:
4. Disable the monman process from starting by modifying the following key in the localopts file:

```
autostart monman = no
```
5. Restart Tivoli Workload Scheduler

Inconsistent time and date in conman and planman output

If you notice inconsistent times and dates in jobs and job streams on an AIX master domain manager, ensure that the system time zone is set correctly. For example, you might notice this problem in the job schedtime or start time, or in other properties related to date and time.

Cause and solution:

The problem might be due to an incorrect setting of the time zone. To set the correct time zone, perform the following steps on the AIX master domain manager:

1. Start smit (System Management Interface Tool).
2. Select System Environments > Change / Show Date, Time, and Time Zone > Change Time Zone Using User Entered Values.
3. Set the relevant time zone. For example, to set the Central European Time (CET) time zone, enter the following values:

* Standard Time ID(only alphabets)	[CET]
* Standard Time Offset from CUT([+ -]HH:MM:SS)	[-1]
Day Light Savings Time ID(only alphabets)	[CEST]
4. Restart the system to make the change effective.

For information about how to set the time zone, see *Tivoli Workload Scheduler: Administration Guide*. For a description of how the time zone works, see *Tivoli Workload Scheduler: User's Guide and Reference*.

Deleting leftover files after uninstallation is too slow

Deleting leftover Onnnn.hhmm files TWA_installation_directory\TWS\stdlist\yyyy.mm.dd\ after uninstalling Tivoli Workload Scheduler is too slow.

Cause and solution:

This problem is caused by a known Microsoft issue on Windows operating systems. It occurs when you try to delete the Onnnn.hhmm files in TWA_installation_directory\TWS\stdlist\yyyy.mm.dd\ on the Windows system after having uninstalled the master domain manager.

To prevent the problem, remove the *Onnnn.hmmm* files permanently using the Shift-Canc keys instead of using the Delete key or sending the files to the Recycle Bin.

Corrupted special characters in the job log from scripts running on Windows

When you run scripts on Windows systems, any special characters resulting from the commands in the script might not be displayed correctly in the job log. This is a display problem that does not affect the correct run of the job. No workaround is currently available for this problem.

Failover Cluster Command Interface deprecated

The `cluster.exe` command-line tool for Failover Clustering was deprecated for Windows Server 2012 platforms. The commands `Startup_clu.cmd`, `ShutDown_clu.cmd`, and `clusterupg` do not work.

Cause and solution:

This occurs because of the deprecation of the `cluster.exe` command-line tool for Failover Clustering on Windows Server 2012 platforms. To avoid this problem, you must reinstall the deprecated Failover Clustering feature `cluster.exe`.

StartUp shows an error after upgrade

Problem:

After upgrading to version 8.6, StartUp script shows the following error:

```
TWS for UNIX/STARTUP 8.5.1
Licensed Materials - Property of IBM*
5698-WSH
(C) Copyright IBM Corp. 1998, 2012 All rights reserved.
* Trademark of International Business Machines
Program code level: 20120510
Killed
ld.so.1: /export/home/svtUser/TWS/trace/atctl: fatal:
libatrc.so: open failed: No such file or directory
Killed
AWSBH0507I A start command was issued for NC121016.
```

Cause and solution:

During the upgrade to version 8.6, the following configuration files:

- `twc_env.sh`
- `twc_env.csh`
- `jobmanrc`
- `TWSCCLog.properties`
- `StartUp`
- `MakePlan`
- `SwitchPlan`
- `SwitchPlan`
- `CreatePostReports`
- `UpdateStats`
- `ResetPlan`

- Sfinal

are not overwritten but the 8.6 version of above files are installed under tws_home/config directory, and therefore to remove the above error message you must merge manually the two version of the files modifying the file under tws_home directory.

Chapter 9. Troubleshooting dynamic workload scheduling

This section provides information that is useful in identifying and resolving problems with dynamic workload scheduling, including how to tune the job processing rate and how to solve common dynamic scheduling problems.

It includes the following sections:

- “How to tune the rate of job processing”
- “Troubleshooting common problems” on page 152
- “Remote command job fails” on page 151

See also *Tivoli Workload Scheduler Administration* in the section on Auditing.

How to tune the rate of job processing

The processing of jobs submitted for dynamic scheduling is handled by the two subcomponents of dynamic workload broker, job dispatcher and resource advisor, through a mechanism of queues and a cache memory. Job dispatcher uses a system of queues into which jobs are placed according to their processing status and thus transmitted to the resource advisor. Resource advisor uses a system of time slots during which it takes a number of jobs from the job dispatcher and allocates them to the resources that will run them.

The `JobDispatcherConfig.properties` and `ResourceAdvisorConfig.properties` configuration files are tuned to suit most environments. However, if your environment requires a high job throughput or if jobs are processed too slowly, you can add the parameters listed below to the specified configuration files and provide customized values. The configuration files are created for dynamic workload broker at installation time and are documented in *IBM Tivoli Workload Scheduler: Administration Guide*.

By default, the parameters listed below are not listed in the configuration files to prevent unwanted modifications. Only expert administrators should set these parameters.

After modifying these parameters, stop and restart dynamic workload broker, as explained in `startbrokerapp`.

JobDispatcherConfig.properties

MaxProcessingWorkers

Job dispatcher queues the submitted jobs according to their processing status. By default the following 3 queues are already specified:

```
Queue.actions.0 = cancel,  
                  cancelAllocation,  
                  completed,  
                  cancelOrphanAllocation  
Queue.actions.1 = execute,  
                  reallocateAllocation  
Queue.size.1     = 20  
Queue.actions.2 = submitted,  
                  notification,  
                  updateFailed
```

Each queue is determined by the keywords:

Queue.actions.queue_number

Specifies the jobs added in this queue based on their processing status. The *queue_number* identifies the queue and ranges from 0 to 9. You can specify a maximum of 10 queues. The following table shows the entire list of process statuses you can specify in the queues.

Table 14. Job processing status to queue jobs for dispatching

Job processing statuses:		
activated	cancel	cancelAllocation
cancelJobCommand	cancelOrphanAllocation	childActivated
childCompleted	childDeactivated	childStarted
completed	deleteJobCommand	execute
getJobLogCommand	getJobPropertiesCommand	holdJobCommand
notification	reallocateAllocation	reconnect
resumeJobCommand	submitJobCommand	submitted
updateFailed	-	-

Unspecified job processing statuses are automatically placed in queue 0.

Queue.size.queue_number

Specifies the number of threads available to the queue identified by *queue_number*. You can specify 1 to 100 threads for each queue you define. The default is the number specified for `MaxProcessingWorkers`.

`MaxProcessingWorkers` specifies the default number of concurrent threads available to each queue. Each job dispatcher queue uses `MaxProcessingWorkers` threads, unless otherwise specified in `Queue.size.queue_number`. The `MaxProcessingWorkers` default is 10. Of the three default queues shown above, only queue 1 has its size specified to 20 threads (or workers). Queues 0 and 2 use the default defined in `MaxProcessingWorkers` (10 threads).

For example, in a test scenario with 250K jobs submitted through the workload broker workstation, the job allocation queues are re-configured as follows:

```
# Override default settings
Queue.actions.0 = cancel,
                 cancelAllocation,
                 cancelOrphanAllocation
Queue.size.0    = 10
Queue.actions.1 = reallocateAllocation
Queue.size.1    = 10
Queue.actions.2 = updateFailed
Queue.size.2    = 10

# Relevant to jobs submitted from
# workload broker workstation, when successful
Queue.actions.3 = completed
Queue.size.3    = 50
Queue.actions.4 = execute
Queue.size.4    = 50
Queue.actions.5 = submitted
Queue.size.5    = 50
```

```
Queue.actions.6 = notification
Queue.size.6    = 50
```

```
# Default for every queue size
MaxProcessingWorkers = 10
```

Tune this parameter carefully to avoid impairing product performance.

HistoryDataChunk

Specifies the number of jobs to be processed at the same time when moving job data to the archive database. This is applicable only to a DB2 RDBMS. This parameter prevents an overload on the job dispatcher. The unit of measurement is jobs. The default value is 1000 jobs.

ResourceAdvisorConfig.properties

MaxAllocsPerTimeSlot

Specifies the number of requests for job allocation to be processed for each time slot. The default value is 100 requests per time slot. By default, each time slot lasts 15 seconds. Increasing this number causes the resource advisor to process a higher number of resource allocation requests per time slot with consequent processor time usage. This also allows the processing of a higher number of jobs per time slot. Decreasing this number causes the resource advisor to process a lower number of resource allocation requests per time slot resulting in a smoother processor usage and slower job submission processing. You can also modify the time slot duration using the `TimeSlotLength` parameter available in this file.

MaxAllocsInCache

Specifies the number of requests for job allocation submitted by job manager to the resource advisor and stored in its cache. This number should be substantially higher than the value specified in the `MaxAllocsPerTimeSlot` parameter. The default value is 5000 allocation requests. Increasing this number causes the resource advisor to process a potentially higher number of resource reservations per time slot with consequent processor time usage. This also allows the processing of a higher number of jobs. Decreasing this number causes the resource advisor to process a lower number of resource reservations per time slot resulting in lower processor usage and slower job submission processing. For optimal performance, this value should be at least 10 times the value specified in the `MaxAllocsPerTimeSlot` parameter.

Remote command job fails

You define and run a remote command job that performs a task on a remote Windows system. If the remote command job goes into ABEND state, and the job log contains a message like: **AWKRCE012E Could not establish a connection to "nc112134.romelab.it.ibm.com" target machine.**, see the following cause and solution.

Cause and solution:

A necessary Windows service might be stopped. Start the Remote Registry Windows service on the remote system.

On Windows 2012 interactive jobs are not visible on dynamic agents

On Windows 2012 you are running an interactive job on a dynamic agent, but no user interface is displayed.

Cause and solution:

Ensure that the Interactive Services Detection service is running on the dynamic agent, and run the interactive job again.

Troubleshooting common problems

The following problems could be encountered with dynamic workload broker:

- “Dynamic workload broker cannot run after the Tivoli Workload Scheduler database is stopped” on page 153
- “Getting an OutofMemory exception when submitting a job” on page 153
- “Getting an error exception when submitting a job on a fresh agent” on page 153

On AIX operating systems the concurrent submission of one hundred or more jobs on the same agent can result in a core dump or in a resource temporarily unavailable message

On AIX operating systems, the concurrent submission of one hundred or more jobs on the same agent can result in a memory dump or in a resource temporarily unavailable message

On AIX operating systems if you submit concurrently one hundred or more jobs on the same agent you can receive a core memory dump or the following message: resource temporarily unavailable

Cause and solution:

This problem is due to insufficient memory and the process number per user allocated to run the jobs concurrently. To solve this problem, verify the value of the following configuration settings and change them as follows:

Ulimit settings

The submission of a significant number of Java jobs requires a large amount of memory. Change the value for data, stack, and memory limits according to the number of jobs you want to submit. The submission of a significant number of native jobs requires a high number of file descriptors and processes. Change the values for nofiles and processes according to the number of jobs you want to submit. The following example gives possible setting values to submit 100 jobs concurrently:

```
time(seconds)      unlimited
file(blocks)       2097151
data(kbytes)       131072
stack(kbytes)      32768
memory(kbytes)     32768
coredump(blocks)   2097151
nofiles(descriptors) 4000
threads(per process) unlimited
processes(per user) unlimited
```

Process number per user

To submit a high number of jobs concurrently you must have a high value for the **maxuproc** setting. Use the **lsattr -E -l sys0 -a maxuproc** command

to verify the number of concurrent processes that a user can create. Use the **chdev -l sys0 -a maxuproc=<value>** command to change the value for the **maxuproc** setting. For example, to submit 100 jobs concurrently use the following command:

```
chdev -l sys0 -a maxuproc=500
```

Dynamic workload broker cannot run after the Tivoli Workload Scheduler database is stopped

Dynamic workload broker cannot run as long as the database is down. When the database is up and running again, restart dynamic workload broker manually with the **startBrokerApplication** command. The command is described in *IBM Tivoli Workload Scheduler: Administration Guide*.

Getting an OutofMemory exception when submitting a job

If you get the following message after you submit a job for dynamic scheduling:

The job with ID *job ID* failed to start.
The error is "unable to create new native thread".

you must tune a property of the scheduling agent.

The property is named **ExecutorsMinThreads** and is located in the **JobManager.ini** file on the agent (for the path, see “Where products and components are installed” on page 1). Its default value is 38 but if this error occurs, you must decrease it to reduce the number of threads created when the job is launched.

The **JobManager.ini** file is described in the *IBM Tivoli Workload Scheduler: Administration Guide*.

Getting an error exception when submitting a job on a fresh agent

If you register a dynamic agent after the generation of a plan and you submit a job or a jobstream on a workstation class using this agent, you get an error message.

This behaviour is normal considering that the agent is not in the plan yet. To avoid getting this error message, do not use agents registered after the plan has been generated.

Chapter 10. Troubleshooting Dynamic Workload Console problems

Describes how to troubleshoot problems with the Dynamic Workload Console related to connections, performance, user access, reports, and others.

This section describes the problems which could occur while using the Dynamic Workload Console:

Note: For the troubleshooting run time scenarios impacting the Tivoli dynamic workload broker environment, refer to the *Tivoli Dynamic Workload Broker: Installation and Configuration guide*

The problems are described in these groups:

- “Troubleshooting connection problems”
- “Troubleshooting performance problems” on page 166
- “Troubleshooting user access problems” on page 168
- “Troubleshooting problems with reports” on page 170
- “Troubleshooting problems with graphical views” on page 177
- “Troubleshooting problems with database” on page 179
- “Troubleshooting other problems” on page 182

Troubleshooting connection problems

The following problems could occur with the connection to the engine or the database:

- “The engine connection does not work” on page 156
- “Test connection takes several minutes before returning failure” on page 157
- “Engine version and connection status not displayed” on page 158
- “Failure in testing a connection or running reports on an engine using an Oracle database” on page 158
- “Connection error when running historical reports or testing connection from an external instance of WebSphere Application Server” on page 159
- “Connection problem with the engine when performing any operation” on page 159
- “Engine connection does not work when connecting to the z/OS connector (versions 8.3.x and 8.5.x)” on page 160
- “Engine connection does not work when connecting to the z/OS connector V8.3.x or a distributed Tivoli Workload Scheduler engine V8.3.x” on page 161
- “Engine connection does not work when connecting to distributed Tivoli Workload Scheduler engine V8.4 FP2 on UNIX” on page 162
- “WebSphere does not start when using an LDAP configuration” on page 163
- “WebSphere Application Server problem when using an LDAP configuration with SSL authentication mechanism” on page 164
- “Engine connection settings are not checked for validity when establishing the connection” on page 165
- “LDAP account locked after one wrong authentication attempt” on page 165

The engine connection does not work

You define an engine connection, you verify that the values entered for the engine connection are correct, and then you click **Test Connection**. The test fails and a connection error message is returned.

Cause and solution:

Assuming that system_A is where you installed the Dynamic Workload Console, and system_B is where you installed Tivoli Workload Scheduler, follow these verification steps to investigate and fix the problem:

1. Verify that there is no firewall between the two systems as follows:
 - a. Make sure the two systems can ping each other. If you are trying to connect to a z/OS engine you must check that the system where the Dynamic Workload Console is installed and the system where the Tivoli Workload Scheduler z/OS connector is installed can ping each other.
 - b. Make sure you can telnet from system_A to system_B using the port number specified in the engine connection settings (for example, 31117 is the default port number for distributed engine).
 - c. Make sure you can telnet from system_A to system_B using the CSIV2 authentication port numbers specified during installation (for example, 31120 is the default server port number and 31121 is the default client port number).

If either of these two steps fails then there might be a firewall preventing the two systems from communicating.

2. Check if you can connect using the **composer** command line interface, or the Dynamic Workload Console to the Tivoli Workload Scheduler engine on system_B using the same credentials specified in the engine connection. If you cannot, then check if the user definition on system_B and the user authorization specified in the Tivoli Workload Scheduler security file are correct.
3. If you are using LDAP or another User Registry on the Dynamic Workload Console make sure that:
 - a. The connection to the user registry works.
 - b. The User Registry settings specified on the Integrated Solutions Console in the **Security** menu under **Secure administration, applications, and infrastructure** are correct.
 - c. You restarted the affected WebSphere Application Server of both the Dynamic Workload Console and Tivoli Workload Scheduler, after configuring the User Registry
 - d. You ran the updateWas and (on Windows) updateWasService scripts after restarting WebSphere Application Server

For more information about how to configure the Dynamic Workload Console to use LDAP or about how to test the connection to a User Registry, refer to the chapter on configuring user security in the *Tivoli Workload Scheduler: Administration Guide*.

4. If you set up to use Single Sign-On between the Dynamic Workload Console and the Tivoli Workload Scheduler engine, make sure you correctly shared the LTPA_keys as described in the chapter on configuring SSL in the *Tivoli Workload Scheduler: Administration Guide*.

Note: Make sure that you correctly shared the LTPA_keys also if you get errors AWSUI0766E and AWSUI0833E. The problem occurs when the realm values are the same for more than one Websphere Application Server (Dynamic Workload

Console, Tivoli Workload Scheduler z/OS connector, or Tivoli Workload Scheduler engine). These steps are usually described only when you configure the Single Sign On, but they are required also when you have the same realm. You have the same realm when you configure all WebSphere Application Servers with the same LDAP user registry and when you install all WebSphere Application Servers on the same machine.

If this checklist does not help you in identifying and fixing your problem then activate tracing on the Dynamic Workload Console by running the steps listed in “Activating and deactivating traces in Dynamic Workload Console” on page 31 (adding also the Java packages `com.ibm.ws.security.*=all:com.ibm.tws.*=all`), and on the Tivoli Workload Scheduler engine by running the following steps:

1. Connect as ROOT to the system where the Tivoli Workload Scheduler engine is located.
2. Edit the file `TWA_home/wastools/TracingProps.properties`, add the statement:
`tws_with_sec=com.ibm.ws.security.*=all:com.ibm.tws.*=all`

and then save your changes.

3. Run the following script to start tracing:

```
<TWA_home>/wastools/changeTraceProperties.sh -user  
<TWS_user> -password <TWS_user_password> -mode tws_with_sec
```

where `<TWS_user>` and `<TWS_user_password>` are the credentials of the Tivoli Workload Scheduler administrator.

Connect to the Dynamic Workload Console again, test the connection to the Tivoli Workload Scheduler engine, and then check the information stored in the following trace logs:

- On the Dynamic Workload Console:

```
<JazzSM_profile_dir>/logs/server1/trace.log
```

where `<JazzSM_profile_dir>` is

On Windows operating systems

`C:\Program Files\IBM\JazzSM\profile`

On UNIX operating systems

`/opt/IBM/JazzSM/profile`

On the Tivoli Workload Scheduler engine:

```
<WAS_profile_path>/logs/server1/trace.log
```

where the default path for `<WAS_profile_path>` is `<TWA_home>/WAS/TWSprofile`

In these files you see the information about the error that occurred. If useful, compare the connection information stored in the traces with the information set for WebSphere Application Server security on both sides. Refer to the *Tivoli Workload Scheduler: Administration Guide* to list the information about the security properties.

Test connection takes several minutes before returning failure

You select an engine connection and click on **Test Connection** to check that the connection is working. The test takes several minutes to complete and then returns a failure.

Cause and solution:

When the **Test Connection** is run, the result is returned only after the timeout expires. The timeout for running the **Test Connection** operation cannot be customized. The connection failed because of one of the following reasons:

- The system where the Tivoli Workload Scheduler engine is installed is not active.
- The IP address or the hostname of the system where the Tivoli Workload Scheduler engine is installed was not correctly specified (in other words, the host name specified by the `showHostProperties` command must be capable of being contacted by the Dynamic Workload Console and vice versa)
- A network firewall prevents the system where the Dynamic Workload Console is installed and the system where the Tivoli Workload Scheduler engine is installed from communicating.

Check which of these reasons causes the communication failure, fix the problem, and then retry.

Engine version and connection status not displayed

The table listing your engine connections, in the Manage Engines panel of the Dynamic Workload Console does not display the engine version or the icon identifying the connection status.

Cause and solution:

Probably it happened because you tried to establish an engine connection using wrong credentials, then you entered the correct credentials and tested the connection again without checking the **Save** option. The correct credentials have not been saved and even though the connection is successful, data relating engine version and connection status is not loaded in the table.

Refresh the panel to make data displayed.

Failure in testing a connection or running reports on an engine using an Oracle database

You test the connection to an engine by specifying the user credentials for an Oracle database, or you run a report on that engine connection. The operation fails and the following error message is displayed:

```
AWSUI0360E The JDBC URL is not configured on the selected engine,
so the reporting capabilities cannot be used.
      Contact the Tivoli Workload Scheduler administrator."
```

Cause and solution:

Make sure that the Tivoli Workload Scheduler administrator has updated the `TWSConfig.properties` file by adding the following key:

```
com.ibm.tws.webui.oracleJdbcURL
```

specifying the JDBC Oracle URL. For example:

```
com.ibm.tws.webui.oracleJdbcURL=jdbc:oracle:thin:@//9.132.235.7:1521/orcl
```

Rerun the operation after the `TWSConfig.properties` has been updated. For more information about showing and changing database security properties for Tivoli Workload Scheduler, refer to the *IBM Tivoli Workload Scheduler: Administration and Troubleshooting guide*.

Connection error when running historical reports or testing connection from an external instance of WebSphere Application Server

You try to test the connection to an engine where you Enable Reporting, or you try to run a historical report, the report fails and the following database connection error is saved to the WebSphere Application Server logs:

```
[date_and_time] 00000044 SystemErr R Exception in thread "WnTransactionThread-10"
java.lang.VerifyError:
class loading constraint violated (class: com/ibm/db2/jcc/c/p method:
getSQLJLogWriter()Lcom/ibm/db2/jcc/SQLJLogWriter;) at pc: 0
[date_and_time] 00000044 SystemErr R at java.lang.J9VMInternals.verifyImpl
(Native Method)
[date_and_time] 00000044 SystemErr R at java.lang.J9VMInternals.verify
(J9VMInternals.java:59)
[date_and_time] 00000044 SystemErr R at java.lang.J9VMInternals.verify
(J9VMInternals.java:57)
[date_and_time] 00000044 SystemErr R at java.lang.J9VMInternals.initialize
(J9VMInternals.java:120)
[date_and_time] 00000044 SystemErr R at com.ibm.db2.jcc.DB2Driver.connect
(DB2Driver.java:163)
[date_and_time] 00000044 SystemErr R at java.sql.DriverManager.getConnection
(DriverManager.java:562)
[date_and_time] 00000044 SystemErr R at java.sql.DriverManager.getConnection
(DriverManager.java:186)
[date_and_time] 00000044 SystemErr R at
```

The Dynamic Workload Console is installed on an external WebSphere Application Server together with other products using either DB2 or Oracle databases.

Cause and solution:

Because of a current WebSphere Application Server limitation, you must run these steps to run historical reports if your Dynamic Workload Console is installed on an external WebSphere Application Server together with other products using either DB2 or Oracle databases.

1. Stop the WebSphere Application Server.
2. Access the directory:
`<TWA_home>/WAS/systemApps/isclite.ear/TWSWebUI.war/WEB-INF/lib`
3. Remove the following JDBC driver files:
`db2jcc.jar`
`db2jcc_license_cu.jar`
`ojdbc14.jar`
4. Start WebSphere Application Server.

Note: This WebSphere Application Server limitation does not affect your activities if:

- You run Actual Production Details and Planned Production Details reports.
- You run operations that do not require to select Enable Reporting in the engine connection properties.

Connection problem with the engine when performing any operation

Whatever operation you try to run in the Dynamic Workload Console, you get an error message saying that there is a connection problem with the engine.

Cause and solution:

Do the following steps:

1. Exit the Dynamic Workload Console.
2. Restart the WebSphere Application Server.
3. Log in again to the Dynamic Workload Console.

Continue with your activities on Dynamic Workload Console.

Engine connection does not work when connecting to the z/OS connector (versions 8.3.x and 8.5.x)

If one of the following errors occurs when running the test connection, follow the steps described in the cause and solution section:

1. AWSUI0766E Test connection to *myengine*: failed. AWSUI0833E The operation did not complete. There was a communication failure. The internal message is: AWSJZC093E The requested engine zserver is not defined.
2. AWSUI0766E Test connection to *myengine* : failed. AWSUI0833E The operation did not complete. There was a communication failure. The internal message is: A communication failure occurred while attempting to obtain an initial context with the provider URL: "corbaloc:iiop:ZOS_CONNECTOR_HOSTNAME:31127".
3. AWSUI0766E Test connection to *myengine* : failed. AWSUI0833E The operation did complete. There was a communication failure. The internal message is: EQQPH26E TME user ID missing in TME user to RACF userid mapping table: myuser@hostname1.test.com

Cause and solution:

The possible causes for the case above are:

1. The name of the server startup job on host side must be defined on the z/OS connector before you perform the test connection from the TDWC.
2. The Websphere Bootstrap port is incorrect. Make sure that any bootstrap address information in the URL is correct and that the target name server is running. A bootstrap address with no port specification defaults to port 2809. Possible causes other than an incorrect bootstrap address or unavailable name server include the network environment and workstation network configuration.
3. The RACF[®] user ID has not been defined in the mapping table on host side.

You can solve the problem as follows:

Environment description example

The environment is composed of a z/OS connector installed on the hostname1.test.com, a TDWC installed on either the same or another system, and a z/OS engine installed on the hostname2.test.com(port 445).

Steps on the z/OS connector side

Define a connection from the z/OS connector to the host side by running the following script located in the directory <ZCONN_INST_DIR>/wastools and then restart WebSphere:

```
> createZosEngine -name zserver -hostName hostname2.test.com/portNumber 445
> stopWas
> startWas
```

where *zserver* is a logical name and can be changed to any other name.

Check the Bootstrap port by running the script showHostProperties.bat (sh) located in the directory <ZCONN_INST_DIR>/wastools.

Steps on the TDWC side

On the TDWC Web interface, define an engine connection from TDWC to the z/OS connector, as follows:

Engine name

Choose any name.

Engine Type

z/OS.

Host Name

Either hostname1.test.com or localhost depending on if TDWC is installed on the same host of Z/CONN or not.

Port Number

The z/OS connector Bootstrap port.

Remote Server Name

zserver (or the name you used in step 2 - createZosEngine).

User ID / Password

For example, the credentials you specified when installing z/OS Connector (that is, the user that owns the z/OS Connector instance). The user can be any user that is authenticated by the User Registry configured on the embedded WebSphere installed with the products.

Note: Bootstrap Port Number in version 8.5.x depends on which product is installed first. If TDWC is installed first, the Bootstrap port is 22809 and subsequent products installed on top of TDWC inherit that. If z/OS Connector is installed first, the Bootstrap port is 31217. If the z/OS connector version is 8.3 FPx, the default Bootstrap port is 31127.

Steps on the z/OS side

Make sure that user myuser@hostname1.test.com is defined in the RACF user ID mapping table on host side (USERMAP parameter in the SERVLOPTS initialization statement).

Engine connection does not work when connecting to the z/OS connector V8.3.x or a distributed Tivoli Workload Scheduler engine V8.3.x

If one of the following errors occurs when running the test connection, follow the steps described in the cause and solution section:

1. AWSUI0766E Test connection to *myengine*: failed. AWSUI0833E The operation did not complete.

```
Reason: AWSJC0005E WebSphere Application Server gives the following error:
CORBA NO_PERMISSION 0x0 No; nested exception is:
org.omg.CORBA.NO_PERMISSION: Trace from server: 1198777258 at host
myhostname.com >>
org.omg.CORBA.NO_PERMISSION: java.rmi.AccessException: ; nested exception is:
com.ibm.websphere.csi.CSIAccessException:
SECJ0053E: Authorization failed for /UNAUTHENTICATED while invoking (Bean)
ejb/com/ibm/tws/zconn/engine/ZConnEngineHome
getEngineInfo(com.ibm.tws.conn.util.Context):
1 securityName: /UNAUTHENTICATED;accessID:
UNAUTHENTICATED is not granted any of the required roles:
TWSAdmin vmcid: 0x0 minor code: 0 completed: No . . .
```

2. AWSUI0778E There was an authentication failure: the user name or password is incorrect.

Cause and solution:

The symptoms above are caused because on the z/OS connector, or on the distributed engine side, the script **webui.sh** (bat) must be run to enable communication with the TDWC. Under the **wastools** directory of the home directory of the installation directory, run these commands:

```
./webui.sh -operation enable -user wasuser  
-password waspwd -port soap_port  
-pwdLTPA anypassword -server server1  
  
./stopWas.sh -user wasuser -password waspwd  
  
./startWas.sh
```

where:

user and password are those specified at installation time.

port is the WebSphere SOAP port (display it by running the command **showHostProperties.sh**).

pwdLTPA is any password used to export and encrypt the LTPA keys.

server is the WebSphere server name. The default is server1.

Engine connection does not work when connecting to distributed Tivoli Workload Scheduler engine V8.4 FP2 on UNIX

If one of the following errors occurs when running the test connection, follow the steps described in the cause and solution section:

AWSUI0766E Test connection to *myengine*: failed.

SECJ0053E: Authorization failed for /UNAUTHENTICATED while invoking (Bean)ejb/com/ibm/tws/conn/engine/ConnEngineHome getEngineInfo (com.ibm.tws.conn.util.Context):1 securityName: /UNAUTHENTICATED;accessID: UNAUTHENTICATED is not granted any of the required roles: TWSAdmin vmcid: 0x0 minor code: 0 completed: No

Cause and solution:

The problem is caused by a missing setting, which is already fixed in later versions of the engine. You can solve the problem by specifying on the engine instance the fully qualified hostname in the **security.xml**. Run the following steps to solve the problem:

1. Stop WebSphere on the engine using the command: `<twc_install_dir>/wastools/stopWas.sh`
2. Back up and then edit the following file (make sure that the editor does not change the formatting): `<JazzSM_profile_dir>/config/cells/JazzSMNode01Cell/nodes/JazzSMNode01/servers/server1/server.xml` where, the default value of `<JazzSM_profile_dir>` is `/opt/IBM/JazzSM/profile`
3. Locate the line related to the CustomUserRegistry, for example:

```
<userRegistries xmi:type="security:CustomUserRegistry"
xmi:id="CustomUserRegistry_1203516338790"
serverId="mywasadmin" serverPassword="{xor}Mj46LCstMA==" limit="0"
ignoreCase="true" useRegistryServerId="true" realm=""
customRegistryClassName="com.ibm.tws.pam.security.registry.
PamUnixRegistryImpl"/>
```

4. Add the fully qualified hostname to the realm attribute, as in the following example:

```
<userRegistries xmi:type="security:CustomUserRegistry"
xmi:id="CustomUserRegistry_1203516338790"
serverId="a840" serverPassword="{xor}Mj46LCstMA==" limit="0"
ignoreCase="true" useRegistryServerId="true"
realm="nc114040.romelab.it.ibm.com"
customRegistryClassName="com.ibm.tws.pam.security.registry.
PamUnixRegistryImpl"/>
```

5. Restart WebSphere on the engine using the command: `<twm_install_dir>/wastools/startWas.sh`

Note: If you have any problems when restarting WebSphere, restore the original `security.xml` and start again.

WebSphere does not start when using an LDAP configuration

The WebSphere startup fails and the `SystemOut.log` file contains one of the following messages with exceptions.

1.

```
SECJ0419I: The user registry is currently connected to the LDAP server
ldap://nc125088.romelab.it.ibm.com:389.
....
WSVR0009E: Error occurred during startup
com.ibm.ws.exception.RuntimeError: com.ibm.ws.exception.RuntimeError:
javax.naming.NameNotFoundException: [LDAP: error code 32 - No Such Object];
remaining name 'ou=asiapacific,dc=test,dc=it'
at com.ibm.ws.runtime.WsServerImpl.bootServerContainer(WsServerImpl.java:199)
at com.ibm.ws.runtime.WsServerImpl.start(WsServerImpl.java:140)
. . .
```
2.

```
SECJ0418I: Cannot connect to the LDAP server ldap://nc125088.romelab.it.
ibm.com:389.....
WSVR0009E: Error occurred during startup
com.ibm.ws.exception.RuntimeError: com.ibm.ws.exception.RuntimeError:
javax.naming.AuthenticationException: [LDAP: error code 49 - 80090308:
LdapErr: DSID-0C090334, comment: AcceptSecurityContext error, data 525,
vece...
```
3.

```
SECJ0270E: Failed to get actual credentials.
The exception is com.ibm.websphere.security.PasswordCheckFailedException:
No user AMusr1@test.it found
at com.ibm.ws.security.registry.ldap.LdapRegistryImpl.checkPassword
(LdapRegistryImpl.java:311)
at com.ibm.ws.security.registry.UserRegistryImpl.checkPassword
(UserRegistryImpl.java:308)
at com.ibm.ws.security.ltpa.LTPAServerObject.authenticate
(LTPAServerObject.java:766)
```
4.

```
SECJ0352E: Could not get the users matching the pattern AMusr1@test.it
because of the following exception javax.naming.CommunicationException:
nc1250881.romelab.it.ibm.com:389 [Root exception is
java.net.UnknownHostException:
nc1250881.romelab.it.ibm.com]
```

Cause and solution:

The answers to the problems are listed below. The answers refer to some of the security properties provided to the wastool script **changeSecurityProperties.sh** (bat).

1. Connect with an LDAP Browser to the LDAP server and verify that the LDAPBaseDN value is a valid Base Distinguished Name and ensure that the LDAPServerId value is an existing user for the LDAPBaseDN.
2. Ask the LDAP administrator for the user and password to perform LDAP queries and set them in the LDAPBindDN or LDAPBindPassword properties.
3. Connect with an LDAP Browser to the LDAP server and verify that the properties of a valid user match the properties specified in the LDAPUserFilter, and also ensure that these properties are congruent with the type of the value specified on the LDAPServerId. For example, the objectCategory must be an existing objectClass and if LDAPServerId is an email address value, then the property to use on the filter must be the "mail" coherently. A valid user filter for the example is: (&(mail=%v)(objectCategory=user)).
4. Ensure that the *LDAPHostName* is a valid existing host and that it can be reached on the network. A useful test is to try to telnet to that host on the *LDAPPort* specified.

After changing the properties as suggested in the above list, run the **changeSecurityProperties.sh** (bat) script again, providing a file containing the updated security properties. Then start WebSphere.

WebSphere Application Server problem when using an LDAP configuration with SSL authentication mechanism

On AIX 6.1 workstations you installed Dynamic Workload Console version 8.6 integrated with a Tivoli Workload Scheduler version 8.6. The WebSphere Application Server uses LDAP with Microsoft active Directory configured in SSL.

The SSL connection between WebSphere Application Server and the Microsoft active Directory LDAP server fails with the following error:

The extended error message from the SSL handshake exception is:
"PKIX path validation failed:
java.security.cert.CertPathValidatorException: The revocation
status of
the certificate with subject (CN=xxx.xx.net, OU=aaaaa,
O=bbbbbb, L=ccccc, ST=ccccc, C=US) could not be determined."

Cause and solution:

The problem is that WebSphere Application Server version 7.0 does not correctly manage the public key infrastructure (PKIX). An SSL Handshake exception is thrown when the WebSphere Application Server starts. Change the *trust manager* settings by performing the following steps:

1. Stop the WebSphere Application Server.
2. Open the file `TWA_home/ewAS/profiles/TIPProfile/config/cells/TIPCell/security.xml` and locate the following line:
`trustManager="TrustManager_TIPNode_2 TrustManager_TIPNode_1"`
3. Change the value of *trustManager* as follows:
`trustManager="TrustManager_TIPNode_1"`

and save the `security.xml` file.

4. Start the WebSphere Application Server.

Engine connection settings are not checked for validity when establishing the connection

You incorrectly defined an engine connection to a distributed engine specifying a value for **Remote Server Name**. The **Remote Server Name** is not a valid setting for a connection to a distributed engine.

The check runs when you save the engine connection definition or when you run a test connection to that engine, but no exception about the incorrect setting is returned.

Cause and solution:

Whenever the test connection is run, only the mandatory fields for that specific type of engine, distributed rather than z/OS, are used to test the connection. Fields that are not mandatory, such as **Remote Server Name** for distributed engine connections are not taken into account.

LDAP account locked after one wrong authentication attempt

LDAP accounts might be blocked even after only one login attempt when connecting using the web user interface or job scheduling console through LDAP/AD authentication, if wrong credentials are provided because of internal LDAP/AD security policy. This happens because one login attempt with wrong credentials using the web user interface or job scheduling console, is transformed into several authentication requests to LDAP.

Cause and solution:

When a single LDAP hostname is mapped to multiple IP addresses in a network configuration, if an invalid password is entered during the login, WebSphere makes as many login attempts as the number of associated IP addresses plus 1. If the resulting number exceeds the maximum number of failed logins allowed by local LDAP/AD security policy, the LDAP account is blocked. In the log file SystemOut.log the following error shows an authentication error because of wrong credentials:

```
ECJ0369E: Authentication failed when using LTPA. The exception is  
javax.naming.AuthenticationException: [LDAP: error code 49 -  
80090308: LdapErr: DSID-0C090334, comment:  
AcceptSecurityContext error, data 52e, vece
```

The WebSphere APAR PK42672 addresses this problem in the following way:

Two new custom properties are available to prevent this issue; use the one suitable for your LDAP configuration:

1. If LDAP is configured using the **wsadmin** command to register backend LDAP server hostnames, in the administration console click **Security > User Registries > LDAP > Custom Properties** and set to true the property `com.ibm.websphere.security.ldap.retryBind`. If this property is set to false, the Application Server does not retry LDAP bind calls. The default value is true.
2. If LDAP is configured associating a hostname with multiple IP addresses using the network configuration, in the administration console click **Security > User Registries > LDAP > Custom Properties** and set to false the property

com.ibm.websphere.security.registry.ldap.singleLDAP. If this property is set to true, the Application Server does not resolve an LDAP hostname to multiple IP addresses. The default value is false.

Troubleshooting performance problems

- “With a distributed engine the responsiveness decreases overtime”
- “Running production details reports might overload the distributed engine”
- “A "java.net.SocketTimeoutException" received”
- “Graphical view and dashboard stop responding on shared WebSphere Application Server on Solaris” on page 178

With a distributed engine the responsiveness decreases overtime

When working with a distributed engine the responsiveness decreases overtime.

Cause and solution:

The problem might be related to multiple production plan report request running on that Tivoli Workload Scheduler engine, since those operations are CPU consuming. Ensure to wait until the report completion before running again other requests of the same kind.

Running production details reports might overload the distributed engine

The WebSphere Application Server on the distributed engine where the production details reports run, is overloaded and the temporary directory is filled up.

Cause and solution:

The amount of memory used by the WebSphere Application Server to extract the data varies depending on the number of objects to be extracted. For example, to extract 70 000 objects required almost 1 GB of RAM. To change the WebSphere Application Server heap size, see Increasing application server heap size.

Note: As a general recommendation, use filters to avoid extracting huge production report files.

A "java.net.SocketTimeoutException" received

You are accessing the Dynamic Workload Console with Internet Explorer 6.0, service pack 2, on a slow workstation (for example: Pentium 4, CPU 1.8 GHz) and are performing one of the following actions, which does not complete:

- You are querying objects in the plan, but on navigating through the result pages the browser hangs while drawing a result page, leaving the page with just the table header and footer shown and none of the result rows displayed. The hang of the browser can be resolved by clicking a button or link, but the missing data is not displayed.
- You are performing either a **Save**, **Edit**, or **Search** operation in the Workload Designer, which hangs for about 60 seconds and then displays one of these two error messages:

AWSUI6171E The operation could not be completed because the Tivoli Dynamic Workload Console server is unreachable. Possible causes are that the Tivoli Dynamic Workload Console server has been stopped or that your login authentication has expired or has become invalid.

AWSUI6182E The operation could not be completed because an internal error occurred. The internal error is: the service name has not been provided.

Cause and solution:

What exactly causes the problem has not been ascertained (it might be a bug in Internet Explorer), but it can be resolved by increasing the value of one of the configurable timeouts in the application server.

Do the following:

1. Identify the instance of WebSphere Application Server running the Dynamic Workload Console where this workstation normally connects to (if it connects to more than one, perform the procedure for all of them)
2. On that instance, edit the WebSphere Application Server configuration file "server.xml". The default location is

```
JazzSM_profile_dir/config/cells/JazzSMNode01Cell/nodes/JazzSMNode01/servers/server1/server.xml
```

where, the default value of *JazzSM_profile_dir* is:

On Windows operating systems

C:\Program Files\IBM\JazzSM\profile

On UNIX operating systems

/opt/IBM/JazzSM/profile

3. Increase the value of the persistentTimeout of the HTTPInboundChannel related to the WCInboundAdminSecure chain section of the file. The default value is 30, but for the given example (Pentium IV, CPU 1.8 GHz) a suggested value to set is 120. An example using the relevant parts of a modified server.xml is as follows:

- a. Identify the WCInboundAdminSecure chain by looking in the chains section:

```
<chains
  xmi:id="Chain_1226491023533"
  name="WCInboundAdminSecure"
  enable="true"
  transportChannels="TCPInboundChannel_1226491023530
                    SSLInboundChannel_1226491023530
                    HTTPInboundChannel_1226491023531
                    WebContainerInboundChannel_1226491023531"/>
```

Note the value of the HTTPInboundChannel.

- b. Use the value of the HTTPInboundChannel to locate its entry:

```
<:transportChannels
  xmi:type="channelService.channels:HTTPInboundChannel"
  xmi:id="HTTPInboundChannel_1226491023531"
  name="HTTP_3"
  discriminationWeight="10"
  maximumPersistentRequests="100"
  keepAlive="true"
  readTimeout="60"
  writeTimeout="60"
  persistentTimeout="120"
  enableLogging="false"/>
```


Modify `persistentTimeout` as has already been done here.

4. Stop the instance of WebSphere Application Server using the **stopWas** command.
5. If a Tivoli Workload Scheduler component is also running under the same instance of the WebSphere Application Server, you need take no further action, as `appservman` will automatically restart the application server. Otherwise, use the **startWas** command.
6. Test the modified value to see if it has resolved the problem. If not, repeat the operation with a larger value, until the problem is resolved.

Troubleshooting user access problems

- “Wrong user logged in when using multiple accesses from the same system”
- “Unexpected user login request after having configured to use Single Sign-On”
- “Authentication problem: the administrative user cannot see the portfolio” on page 169
- “Single Sign-On authentication doesn't work anymore from Dynamic Workload Console with master version 8.5.1 and master upgraded to version 8.6.0.” on page 169
- “Authentication problem when opening the Workload Designer” on page 170

Wrong user logged in when using multiple accesses from the same system

You try to access the Dynamic Workload Console as *user2* using Firefox or Internet Explorer 7, where a connection as *user1* is already active in the same browser. In the case of Firefox the problem occurs if *user1* is active in *any* other Firefox window or tab. In Internet Explorer 7 the problem only occurs if the other user is active in a different tab of the same browser instance. But in both cases the result is the same: the browser logs you in to the Dynamic Workload Console as *user1* instead of *user2*.

Cause and solution:

This is a browser limitation. If you have an active connection through Internet Explorer 7 to the Dynamic Workload Console, and you want to open another session on the same system, you need only to open a different browser window. If the active connection is on Firefox, however, you must use a different browser. For a list of supported browsers, see the Dynamic Workload Console System Requirements Document at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27038329>.

Unexpected user login request after having configured to use Single Sign-On

It might happen that, after running successfully all the steps required to configure the Single Sign-On between the Dynamic Workload Console and a Tivoli Workload Scheduler engine, when you try to test the connection or run a task on that engine, you are unexpectedly prompted to enter your user credentials to connect. This behavior means that the Single Sign-On method is not working properly on that engine and you receive the following exception:

Cause and solution:

Make sure that the *application_server/profiles/profile_name/config/cells/cell_name/security.xml* files of both the Dynamic Workload Console and the Tivoli Workload Scheduler engine have identical values assigned to the *realm* field of the *security:LDAPUserRegistry* section. This setting belongs to the WebSphere Application Server profile configuration.

For example, even though you ran all the required steps to configure the Single Sign-On, it might still not work if you set *realm="myHost.myDomain:389"* on the Dynamic Workload Console and *realm="myHost:389"* on the Tivoli Workload Scheduler engine. To make it work, you must enter:
realm="myLDAPHost.myDomain:389".

After setting the correct string, click **Apply**, save the configuration and restart the Dynamic Workload Console and WebSphere Application Server.

Single Sign-On authentication doesn't work anymore from Dynamic Workload Console with master version 8.5.1 and master upgraded to version 8.6.0.

You have a Single Sign-On configuration between the Dynamic Workload Console version 8.6.0 or fix pack 1 and a Tivoli Workload Scheduler engine earlier version with federated LDAP repository name: *ou=svt,dc=test,dc=it*. After the Tivoli Workload Scheduler upgrade to version 8.6.0, when you try to test the connection or run a task on that engine from the Dynamic Workload Console, you have the following message:

```
[3/27/12 12:28:54:499 CEST] 00000009 ContextManage 3
Setting recSavedSubject:Subject:
Principal: nc125088.romelab.it.ibm.com:389/fcarteri1
Public Credential: com.ibm.ws.security.auth.WSCredentialImpl@64aa64aa
Private Credential: com.ibm.ws.security.token.SingleSignonTokenImpl@1de21de2
Private Credential: com.ibm.ws.security.token.AuthenticationTokenImpl@46694669
Private Credential: com.ibm.ws.security.token.AuthorizationTokenImpl@17c917c9
```

```
[3/27/12 12:28:54:499 CEST] 00000009 ContextManage < setSavedSubjects Exit
[3/27/12 12:28:54:453 CEST] 0000000b ContextManage < runAs(System)
-> Exception occurred. Exit
```

```
com.ibm.websphere.wim.exception.InvalidUniqueNameException:
CWWIM0515E The 'CN=dcore,OU=SVT,DC=test,DC=it' entity is not in
the scope of the 'defined' realm.
```

Cause and solution:

The upgrade process changes the security configuration with a federated repository named *o=twalLDAP* instead of the old value. This causes that SSO authentication doesn't work from the Dynamic Workload Console to the engine. To solve the problem, see the Technote: <http://www-01.ibm.com/support/docview.wss?uid=swg21592036>.

Authentication problem: the administrative user cannot see the portfolio

You log in to the Dynamic Workload Console using the administrative user but neither the Dynamic Workload Console entries, nor the Dashboard Application Services Hub security entries are displayed in the navigation tree.

Cause and solution:

Probably, during manual configuration of the authentication mechanism to access the Dynamic Workload Console, the old primary admin ID (twapam) entry was deleted from the repository before all the configuration steps had been completed.

Perform the following procedure to solve the authentication problem:

1. Assign the following roles to a user in the repository:
 - ISC Admin
 - Admin security manager
2. Log into the Dynamic Workload Console using this user and assign the following roles to the primary administrative user (the new administrative user):
 - Iscadmins
 - TDWBAdministrator
 - TWSWEBUIAdministrator
3. Use the new administrative user to revoke *ISC Admin* and *Admin security manager* permissions from the user chosen in 1 or, optionally, to delete the user.

Authentication problem when opening the Workload Designer

You try to open the Workload Designer but the action fails and you receive a message saying that the engine credentials are wrong, even though you entered the correct user name and password. This happens in a configuration with the Dynamic Workload Console connected to Tivoli Workload Scheduler for z/OS connector on z/OS WebSphere Application Server.

The problem is due to the definition of the engine connection that contain the engine IP address instead of the hostname.

To solve the problem, open the Engine Connection Properties panel and replace the IP address with the hostname in the Connection Data section.

Troubleshooting problems with reports

- “The output of a report run on Job Statistics View shows -1 in the Average CPU Time and Average Duration fields” on page 171
- “The validate command running on a custom SQL query returns the error message AWSWUI0331E” on page 171
- “The output of report tasks is not displayed in a browser with a toolbar installed” on page 171
- “WSWUI0331E error when running reports on an Oracle database” on page 172
- “CSV report looks corrupted on Microsoft Excel not supporting UTF8” on page 172
- “Insufficient space when running production details reports” on page 172
- “After Tivoli Workload Scheduler upgrades from version 8.3 to version 8.5 some fields in the output of reports show default values (-1, 0, unknown, regular)” on page 173
- “Report error: the specified run period exceeds the historical data time frame” on page 173

The output of a report run on Job Statistics View shows -1 in the Average CPU Time and Average Duration fields

You run a report accessing the Job Statistics Database View, such as Job Run Statistics or a Custom SQL report, and the output shows the value -1 in Average CPU Time and Average Duration fields.

Cause and solution:

The historical report, regardless of what kind of report you run (for Jobs, Workstations, or Custom SQL), reads in the database the information about the previous production plan run. If some fields in a database view are empty, the value returned in the report output is -1. This means that if you run **JNextPlan** for the first time, and then you run for example the Job Run Statistics report, the value of Average CPU Time and Average Duration fields is -1.

Run **JNextPlan** again, or wait for the *final* job stream to run, to populate the database views and get values different from -1.

The validate command running on a custom SQL query returns the error message AWSWUI0331E

You are creating a Custom SQL report, and you run the **Validate** command to check your query. The validate fails and the following error message is returned:

```
AWSWUI0331E The SQL query could not be validated. The database internal message is:
[ibm][db2][jcc][10103][10941] Method executeQuery cannot be used for update.
```

Cause and solution:

The validate failure is caused by a syntax error in the query statement, for example, a typing error, such as:

```
sele Workstation_name,Job_name,Job_start_time from MDL.JOB_HISTORY_V
where Workstation_name like 'H%'
```

In this query, *sele* is written in place of *select*.

Verify the SQL query is correct and, optionally, try to run the same query from the DB2 command line to get additional details.

The output of report tasks is not displayed in a browser with a toolbar installed

You tested that the connection to the database set in the engine connection works properly but, after you run a report task, no window opens in your browser to display the task results. You have a third-party toolbar installed on your browser.

Cause and solution:

A third-party toolbar (such as Yahoo! or Google or similar) installed on top of the browser might conflict with the correct operation of the Dynamic Workload Console reporting feature. To make the reporting feature work correctly you must uninstall the toolbar and then rerun the report task.

WSWUI0331E error when running reports on an Oracle database

You try to run a report on an engine connection where an Oracle database has been referenced. The report task fails and the following error is displayed:

WSWUI0331E SQL validate failure.The database internal message is:ORA-00942:
table or view does not exist

If you try to run an SQL query statement in the Oracle database on the same table or view using the userid specified for the database connection in the engine connection properties, the query runs successfully.

Cause and solution:

On Oracle databases only, you must run these steps, as Oracle database administrator, to allow the database user specified in the engine connection properties to run reports from the Dynamic Workload Console:

1. Assign to the database user the "CREATE TABLE" Oracle System privilege.
2. Run the following script:

On Windows

```
TWA_home\TWS\dbtools\oracle\script\dbgrant.bat
```

On UNIX:

```
TWA_home/dbtools/oracle/script/dbgrant.sh
```

CSV report looks corrupted on Microsoft Excel not supporting UTF8

You run a report asking to save the result in a CSV file. When you open the CSV file using Microsoft Excel, the content of the file looks corrupted.

Cause and solution:

To bypass this problem, make sure that the version of Microsoft Excel you are using supports the UTF8 character set. If it does not, install a more recent version that supports UTF8. Then, follow these steps to correctly open CSV reports from Microsoft Excel:

1. Open Microsoft Excel.
2. In the **Data** menu entry, select **Import External Data** and then **Import Data**.
3. Select the CSV file saved and click **Open**.
4. In the field File Origin, select **UTF8**.

Insufficient space when running production details reports

When running production details reports the temporary directory on the Tivoli Workload Scheduler engine where the reports run, could be filled up.

Cause and solution:

You need to free some space in the temporary directory on the Tivoli Workload Scheduler engine before continuing to work on that engine.

After Tivoli Workload Scheduler upgrades from version 8.3 to version 8.5 some fields in the output of reports show default values (-1, 0, unknown, regular)

After migrating Tivoli Workload Scheduler from version 8.3 to version 8.5, the output on the Dynamic Workload Console of reports run on old migrated jobs show default values for the new fields introduced since version 8.3.

Cause and solution:

This is not a problem or a limitation but the result of migrating data from old tables to new tables containing newly created fields. After migration, it is necessary to assign a value to the new fields introduced since version 8.3 for job runs that occurred before migrating. The values assigned by default to these new fields are:

For job run statistic reports:

Table 15. Default settings for new job run statistic reports

Value	Field
0	Number of "Long Duration" job runs
0	Number of "Suppressed" job runs
0	Number of "Started Late" job runs
0	Number of "Ended late" job runs
0	Total Reruns
-1	Average CPU Time
-1	Average Duration

For job run history reports:

Table 16. Default settings for new job run history reports

Value	Field
unknown	Workstation Name (Job Stream)
-1	Started Late (delay hh:mm)
-1	Ended Late (delay hh:mm)
-1	Estimated Duration (hh:mm)
No	Long Duration
Regular	Run Type
-1	Iteration Number
0	Return Code
0	Job Number
unknown	Login

Report error: the specified run period exceeds the historical data time frame

You define a report specifying a valid date range as execution period in the filter criteria. When you run the report, you receive the following warning message:

AWSUI2003IThe specified run period exceeds the historical data time frame. The database contains historical data from....

Cause and solution:

This problem occurs if different time zones are used when creating the report and when running it. To solve the problem, edit the report task, change the time zone making it equal to the time zone currently specified in the user preferences and run the report again.

Troubleshooting problems with browsers

- "Default tasks are not converted into the language set in the browser"
- "'Access Error' received when launching a task from the browser bookmark"
- "If you close the browser window, processing threads continue in the background" on page 175
- "Unresponsive script warning with Firefox browser" on page 175
- "Plan View panel seems to freeze with Internet Explorer version 7" on page 175
- "Workload Designer does not show on foreground with Firefox browser" on page 176
- "Some panels in Dynamic Workload Console might not be displayed correctly in Internet Explorer, version 8 and 9" on page 176
- "Web page error with Internet Explorer, version 9" on page 177
- "Dynamic Workload Console problems with Internet Explorer developer tools" on page 177

Default tasks are not converted into the language set in the browser

An existing user logs in to the Dynamic Workload Console using a browser where the language set is different from the language that was set in the browser the first time he logged in. In the Manage Tasks window, the default tasks are not translated into the new language.

Cause and solution:

The default tasks are created, using the current language set in the browser, when the new user logs into the Dynamic Workload Console for the first time. To have the default tasks translated into a different language, the WebSphere Application Server administrator must create a new Dynamic Workload Console user, and use that to login to the Dynamic Workload Console for the first time using a browser configured with the requested language. By doing this the default tasks are created using the requested language.

"Access Error" received when launching a task from the browser bookmark

A Dynamic Workload Console task has been saved in the list of bookmarks of the browser. You try to launch the task using the bookmark but you receive the following error message:

"User does not have access to view this page, use the browser back button to return to previous page."

Cause and solution:

You do not have the necessary role required to run the task. To run a task you must have a role that allows you to access the Dynamic Workload Console panels that are relevant to the type of task you need.

For more information about setting roles to work with the Dynamic Workload Console, see the Administration Guide, under the section about Configuring new users to access Dynamic Workload Console

If you close the browser window, processing threads continue in the background

You perform an action or make a selection and immediately close the browser window. You expect that processing terminated but the messages stored in the `SystemOut.log` file show that processing continued in the background.

Cause and solution:

This is normal behavior for any WEB application, when the client browser is closed no notification is delivered to the server according to the HTTP protocol specifications. This is the reason why the last triggered thread continues to process even after the browser window was closed. You do not need to run any action, just allow the thread to end.

Unresponsive script warning with Firefox browser

When opening the Workload Designer with Firefox, the following warning message might appear:

Warning: Unresponsive script

A script on this page may be busy, or it may have stopped responding.

You can stop the script now, or you can continue to see if the script will complete.

Cause and solution:

This is caused by a Firefox timeout. If prompted with this warning message, choose the "Continue" option.

This behavior of Firefox is ruled by its `dom.max_script_run_time` preference, which determines the timeout that the browser must wait for before issuing the warning. The default value is 10 seconds, and might be changed to another value according to your needs.

To change this value, do the following:

1. Type `about:config` in the address field of the browser.
2. Scroll down to the preference, select it, change the value, and click **OK**.

Plan View panel seems to freeze with Internet Explorer version 7

When using Internet Explorer version 7, some actions performed in sequence might cause the **Plan View** browser window to freeze and stay frozen for about 5 minutes. After this timeframe the browser window resumes.

Cause and solution:

Action sequences that might cause this problem typically include opening multiple **Plan View** panels at the same time and refreshing the **Plan View** panels that were already open.

To avoid or limit this behavior add the Dynamic Workload Console website to the **Local intranet** security zone of Internet Explorer 7, with its default security level.

Blank page displayed (in High availability disaster recovery configuration)

You are in a high availability disaster recovery (HADR) configuration and the Dynamic Workload Console displays blank panels when trying to retrieve information from DB2.

Cause and solution:

When DB2 primary node stops, every Dynamic Workload Console request waits for a manual switch to a standby node.

If you have a HADR DB2 configuration related to your Tivoli Workload Scheduler engine, and you get an empty or blocked panel in the Dynamic Workload Console, verify that your primary node is up and running.

Workload Designer does not show on foreground with Firefox browser

With Firefox, if you open the Workload Designer from a graphical view (with the Open Job definition or the Open Job stream definition commands), and the Workload Designer window is already open, this window might not be moved to the foreground.

Solution:

To fix this problem, change the Firefox settings as follows:

1. On the Firefox action bar select **Tools**, then **Options**, then **Content**, and finally **Advanced**
2. Enable the Raise or lower windows option

Some panels in Dynamic Workload Console might not be displayed correctly in Internet Explorer, version 8 and 9

When using Internet Explorer version 8 or 9, some panels in Dynamic Workload Console, might not display as expected, for example

- The Graphical View or some Dashboard graphics, might not be displayed correctly.
- When duplicating a Monitor task, the entire Dashboard Application Services Hub navigation toolbar appears duplicated.

Cause and solution:

This problem can be due to incorrect settings in Internet Explorer.

To avoid or limit this behavior, use the following workarounds:

- Add the Dynamic Workload Console web site to the **Local intranet** security zone of Internet Explorer, with its default security level.

- 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.
- Turn off Internet Explorer Enhanced Security mode as described in the following documentation: *IBM Tivoli Security Information and Event Manager*, section about Disabling Enhanced Security Configuration.. In fact, Dashboard Application Services Hub does not support Internet Explorer with Enhanced Security mode active.

Web page error with Internet Explorer, version 9

When using Internet Explorer version 9, some panels in Dynamic Workload Console, might not be displayed correctly and the page issues the following web page error: "object Error".

Cause and solution:

To solve this problem, clear the browser cache.

Dynamic Workload Console problems with Internet Explorer developer tools

When using Internet Explorer version 8 or 9, the Dashboard might not display correctly when working with Internet Explorer developer tools open. Some sections of the Dashboard might remain loading without completing the update.

Cause and solution:

This problem is due to a conflict between how events are managed respectively by Internet Explorer developer tools and the Dashboard.

To avoid or limit this behavior, close the developer tools and try again the operation.

Troubleshooting problems with graphical views

- "Language-specific characters are not correctly displayed in graphical views"
- "Plan View limit: maximum five users using the same engine" on page 178
- "Graphical view and dashboard stop responding on shared WebSphere Application Server on Solaris" on page 178
- "Job stream no longer displayed in job stream view" on page 178
- "AWSITA122E or AWKRAA209E error while working with jobs in the Workload Designer" on page 178

Language-specific characters are not correctly displayed in graphical views

When working with the graphical views some language specific characters might not be displayed correctly.

Cause and solution:

This might occur because the necessary language files have not been installed on the computer on which the Dynamic Workload Console is running. To solve the problem, install the operating system language files on the system hosting the Dynamic Workload Console.

Plan View limit: maximum five users using the same engine

If you try to open the **Plan View** when five users are already concurrently using it, with the same engine, your request is rejected with the following error message: AWSJC0136E No more than 5 users are allowed to perform this operation at the same time. The maximum number of concurrent requests has been reached: please try again later.

Cause and solution:

The maximum number of users that can use the **Plan View** connected to the same engine is five.

If needed, you can modify this limit by editing the `com.ibm.tws.conn.plan.view.maxusers` property in the `TWSConfig.properties` file.

Job stream no longer displayed in job stream view

Working with an empty job stream in the job stream view, after you have clicked the **Hide/Show dependencies** button, the job stream is no longer displayed.

This only happens when working with job streams that do not contain any job.

Cause and solution:

The problem is due to IBM ILOG limitation that prevents a correct job stream visualization.

Reload the job stream to solve the problem and have a correct view.

Graphical view and dashboard stop responding on shared WebSphere Application Server on Solaris

On Solaris sparc systems, where the Dynamic Workload Console and the Master Domain Manager share the same WebSphere Application Server when you open some panels, like the graphical view or the dashboard, the console might stop responding and you might receive an error message saying that the server is unreachable. In the log files you could find an `OutOfMemory` exception (`java.lang.OutOfMemoryError: PermGen space`).

Cause and solution:

The problem is caused by insufficient `MaxPermSize`. To solve the problem, increase the **MaxPermSize** to 512 MB in `server.xml` file.

AWSITA122E or AWKRAA209E error while working with jobs in the Workload Designer

While you are working with a job in the Workload Designer, you might receive one of the following errors:

AWSITA122E - A Java exception occurred while calling the command

Cause and solution:

An unexpected error occurred while running a Java method.

In case it happens, check the following log files:

1. Check the JobManager_message.log file.
2. Check the latest /opt/ibm/TWA/TWS/JavaExt/eclipse/configuration/*.log file

AWKRAA209E - The job with advanced options with ID "application_type" was not found.

Cause and solution:

The job with advanced options cannot be found.

In case it happens, perform the following steps:

1. Ensure that the job plug-in is present in the /opt/IBM/TWA/TWS/JavaExt/eclipse/plugins directory.
2. Ensure that the job plug-in is listed in the /opt/IBM/TWA/TWS/JavaExt/eclipse/configuration/config.ini file
3. Check the latest /opt/IBM/TWA/TWS/JavaExt/eclipse/configuration/*.lma log file.

Troubleshooting problems with database

- "Import preferences fails"
- "Communication failure with DB2 when working with the Dynamic Workload Console on a Red Hat Enterprise Linux (RHEL) Version 5.6 system"

Communication failure with DB2 when working with the Dynamic Workload Console on a Red Hat Enterprise Linux (RHEL) Version 5.6 system

When trying to access the user preferences stored on a DB2 repository from a Dynamic Workload Console on a Red Hat Enterprise Linux Version 5.6, you might receive the following message: "Unable to access to preferences repository".

Cause and solution:

There are some compatibility issues between Dynamic Workload Console DB2 driver and RHEL 5.6 that in some cases prevent the Dynamic Workload Console from accessing DB2 repository properly.

To solve this problem, upgrade the RHEL to Red Hat Enterprise Linux Version 6.

Import preferences fails

When trying to import your settings repository from XML file specifying the **Cancel and recreate** option, the import operation fails and you receive the following message: *AWSUI0924E Preferences import operation failed: Unable to create database.* Moreover, the following error is logged into SystemErr.log file: DB2 SQL error: SQLCODE: -601, SQLSTATE: 42710, SQLERRMC: TDWC.TDWC_PREFERENCEABLE;TABLE.

Cause and solution:

This might occur because the database user with administrative authority specified to import the settings does not have the privileges required to drop the existing Dynamic Workload Console tables that were created with Dynamic Workload Console 8.6.0.0.

To solve this problem, provide the specified user also with CONTROL privilege on all Dynamic Workload Console tables.

For example:

```
db2 "GRANT CONTROL ON TABLE TDWC.TDWC_CONFIGURATIONPROPERTY TO USER myuser"
db2 "GRANT CONTROL ON TABLE TDWC.TDWC_CREDENTIAL TO USER myuser"
db2 "GRANT CONTROL ON TABLE TDWC.TDWC_ENGINECONNECTION TO USER myuser"
db2 "GRANT CONTROL ON TABLE TDWC.TDWC_MEQUERTASK TO USER myuser"
db2 "GRANT CONTROL ON TABLE TDWC.TDWC_PREFERENCEABLE TO USER myuser"
db2 "GRANT CONTROL ON TABLE TDWC.TDWC_QUERYTASK TO USER myuser"
db2 "GRANT CONTROL ON TABLE TDWC.TDWC_REPORTTASK TO USER myuser"
```

Troubleshooting configuration problems

The following problems could occur during the dynamic workload Console configuration.

IHS load balancing configuration for Dynamic Workload Console cluster fails

Problem:

When trying to perform the IHS load balancing configuration for Dynamic Workload Console cluster, the configuration fails with the PLGC0028E An exception occurred while reading the deployed applications for the specified server. JazzSMNode01Cell.JazzSMNode01.server1. error:

```
IBM WebSphere Application Server, Release 8.5
WebSphere Plugin Configuration Generator
Copyright IBM Corp., 1997-2012
PLGC0061I: Debug log file = /opt/IBM/JazzSM/profile/logs/plugin-cfg.log
PLGC0013I: The plug-in is generating a server plug-in configuration file for all
of the servers in the cell. JazzSMNode01Cell.
PLGC0041W: The plug-in is unable to load the com.ibm.wkplc.generator.LotusCfgHelper
helper class.
java.lang.IncompatibleClassChangeError: org.objectweb.asm.ClassVisitor
    at java.lang.ClassLoader.defineClassImpl(Native Method)
    at java.lang.ClassLoader.defineClass(ClassLoader.java:286)
    at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:154)
    at java.net.URLClassLoader.defineClass(URLClassLoader.java:711)
    at java.net.URLClassLoader.access$400(URLClassLoader.java:92)
    at java.net.URLClassLoader$ClassFinder.run(URLClassLoader.java:1159)
    at java.security.AccessController.doPrivileged(AccessController.java:314)
    at java.net.URLClassLoader.findClass(URLClassLoader.java:594)
    at com.ibm.ws.bootstrap.ExtClassLoader.findClass(ExtClassLoader.java:204)
    at java.lang.ClassLoader.loadClassHelper(ClassLoader.java:713)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:690)
    at com.ibm.ws.bootstrap.ExtClassLoader.loadClass(ExtClassLoader.java:119)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:660)
    at java.lang.ClassLoader.defineClassImpl(Native Method)
    at java.lang.ClassLoader.defineClass(ClassLoader.java:286)
    at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:154)
    at java.net.URLClassLoader.defineClass(URLClassLoader.java:711)
    at java.net.URLClassLoader.access$400(URLClassLoader.java:92)
    at java.net.URLClassLoader$ClassFinder.run(URLClassLoader.java:1159)
    at java.security.AccessController.doPrivileged(AccessController.java:314)
    at java.net.URLClassLoader.findClass(URLClassLoader.java:594)
    at com.ibm.ws.bootstrap.ExtClassLoader.findClass(ExtClassLoader.java:204)
```

```

    at java.lang.ClassLoader.loadClassHelper(ClassLoader.java:713)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:690)
    at com.ibm.ws.bootstrap.ExtClassLoader.loadClass(ExtClassLoader.java:119)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:660)
    at java.lang.J9VMInternals.verifyImpl(Native Method)
    at java.lang.J9VMInternals.verify(J9VMInternals.java:85)
    at java.lang.J9VMInternals.initialize(J9VMInternals.java:162)
    at com.ibm.ws.amm.commonarchive.AnnotationsProcessorImpl.merge
(AnnotationsProcessorImpl.java:87)
    at com.ibm.ws.amm.commonarchive.AnnotationsProcessorImpl.merge
(AnnotationsProcessorImpl.java:66)
    at org.eclipse.jst.j2ee.commonarchivecore.internal.impl.
WARFileImpl.processAnnotations(WARFileImpl.java:923)
    at com.ibm.ws.webfragmerger.WebFragMergerImpl.merge
(WebFragMergerImpl.java:493)
    at org.eclipse.jst.j2ee.commonarchivecore.internal.impl.WARFileImpl.
mergeAnnotationsAndFragments(WARFileImpl.java:869)
    at org.eclipse.jst.j2ee.commonarchivecore.internal.impl.WARFileImpl.
getDeploymentDescriptor(WARFileImpl.java:648)
    at org.eclipse.jst.j2ee.commonarchivecore.internal.impl.WARFileImpl.
getDeploymentDescriptor(WARFileImpl.java:367)
    at org.eclipse.jst.j2ee.commonarchivecore.internal.impl.WARFileImpl.
getStandardDeploymentDescriptor(WARFileImpl.java:1498)
    at org.eclipse.jst.j2ee.commonarchivecore.internal.impl.EARFileImpl.
getDeploymentDescriptor(EARFileImpl.java:817)
    at org.eclipse.jst.j2ee.commonarchivecore.internal.impl.EARFileImpl.
getDeploymentDescriptor(EARFileImpl.java:789)
    at com.ibm.websphere.plugincfg.generator.ConfigurationParser.
getDeployedModules(ConfigurationParser.java:2503)
    at com.ibm.websphere.plugincfg.generator.ConfigurationParser.
setClusterDeployedModuleInfo(ConfigurationParser.java:2192)
    at com.ibm.websphere.plugincfg.generator.ConfigurationParser.
<init>ConfigurationParser.java:261)at com.ibm.websphere.plugincfg.
generator.PluginConfigGenerator.generate9
    (PluginConfigGenerator.java:798) at com.ibm.websphere.plugincfg.
generator.PluginConfigGenerator.generate
    (PluginConfigGenerator.java:762) at com.ibm.websphere.plugincfg.
generator.PluginConfigGenerator.main
    (PluginConfigGenerator.java:1357) at sun.reflect.
NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke
(NativeMethodAccessorImpl.java:88)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke
(DelegatingMethodAccessorImpl.java:55)
    at java.lang.reflect.Method.invoke(Method.java:613)
    at com.ibm.ws.bootstrap.WSLauncher.main(WSLauncher.java:277)

PLGC0028E: An exception occurred while reading the deployed applications for
the specified server.
JazzSMNode01Cell.JazzSMNode01.server1.
PLGC0005I: Plug-in configuration file =
/opt/IBM/JazzSM/profile/config/cells/plugin-cfg.xml

```

Cause and solution:

The WebSphere Application Server V8.5.0 that is a Dynamic Workload Console prerequisites has problems.

To solve this problem, install the WebSphere Application Server 8.5.0.1-WS-IFPM88913 8.5.1.20130513_1520 fix, as described in “WebSphere Application Server fix installation procedure.”

WebSphere Application Server fix installation procedure

To install the WebSphere Application Server 8.5.0.1-WS-IFPM88913 8.5.1.20130513_1520 fix, perform the following steps:

1. Locate the `<eImage_dir>/IFPM88913` directory, where `<eImage_dir>` is the directory where you downloaded the Tivoli Workload Scheduler or Dynamic Workload Console images.
2. Open the Installation Manager wizard.
3. Add an Installation Manager repository that contains the `8.5.0.1-WS-IFPM88913 8.5.1.20130513_1520` fix in network format, by running the following steps:
 - a. Select **File > Preferences**. The Repositories page is displayed and shows available repositories, repository locations, and the connection status for the repositories.
 - b. Click **Add Repository**. The Add Repository page is displayed.
 - c. In the Add Repository page, click **Browse**, select the `<eImage_dir>/IFPM88913/repository.config` file and click **Open**.
 - d. Click **OK**.
4. Install the WebSphere Application Server fix, by performing the following steps:
 - a. In the Installation Manager wizard, click **Update**.
 - b. In the Fixes panel, select the IBM WebSphere Application Server 8.5.0.1 offering and select the `8.5.0.1-WS-IFPM88913 8.5.1.20130513_1520` fix and click **Next**.
5. Leave the default fields values and click **Next**.
6. Click **Update**.

Troubleshooting other problems

- "The deletion of a workstation fails with the "AWSJOM179E error" on page 99
- "Data not updated after running actions against monitor tasks results" on page 183
- ""Session has become invalid" message received" on page 183
- "Actions running against scheduling objects return empty tables" on page 184
- "Default tasks are not converted into the language set in the browser" on page 174
- ""Access Error" received when launching a task from the browser bookmark" on page 174
- "The validate command running on a custom SQL query returns the error message AWSWUI0331E" on page 171
- "If you close the browser window, processing threads continue in the background" on page 175
- "The list of Available Groups is empty in the Enter Task Information window" on page 185
- "JVM failure when working with the Dynamic Workload Console on a Red Hat Enterprise Linux (RHEL) Version 5 system" on page 186
- "Communication failure with DB2 when working with the Dynamic Workload Console on a Red Hat Enterprise Linux (RHEL) Version 5.6 system" on page 179
- "Missing daylight saving notation in the time zone specification on Dynamic Workload Console 8.4 Fix Pack 1 and later" on page 186
- "A "java.net.SocketTimeoutException" received" on page 186
- "Blank page displayed (in High availability disaster recovery configuration)" on page 176
- "Some panels in Dynamic Workload Console might not be displayed correctly" on page 187
- "Extraneous exception logged in SystemOut" on page 188

- “Filtering task results might not work as expected” on page 188
- “Sorting task results might not work as expected” on page 189

The deletion of a workstation fails with the "AWSJOM179E error

You want to delete a workstation either using **Composer** or the Dynamic Workload Console and the following error occurs:

AWSJOM179E An error occurred deleting definition of the workstation {0}.
The workload broker server is currently unreachable.

Cause and solution:

This problem occurs if you removed a dynamic domain manager without following the procedure that describes how to uninstall a dynamic domain manager in the *Tivoli Workload Scheduler: Planning and Installation Guide*.

To remove workstations connected to the dynamic domain manager, perform the following steps:

1. Verify that the dynamic domain manager was deleted, not just unavailable, otherwise when the dynamic domain manager restarts, you must wait until the workstations register again on the master domain manager before using them.
2. Delete the workstations using the following command:

```
composer del ws <workstation_name>;force
```

Data not updated after running actions against monitor tasks results

After running an action on a list of objects returned from running a monitor task the list is not updated.

Cause and solution:

The scheduling objects lists are not automatically updated after running actions. Click on the **Refresh** button to update the list of objects.

"Session has become invalid" message received

You try to use the Dynamic Workload Console user interface, your working session closes, and you get the following warning:

Session has become invalid

Your session has become invalid. This is due to a session timeout, an administrator has logged you out, or another user has invalidated your session by logging on with the same User ID.

Cause and solution:

Check which reason among those listed in the warning has occurred, solve the issue, and then log in again to continue your working session.

If the session expired because either the HTTP session or the Lightweight Third Party Authentication (LTPA) session timeout was exceeded, you might decide to customize the timeout settings to values that are appropriate for your environment.

For instructions on how to do this, see the topic on session timeout settings in the *Performance* chapter of the *Tivoli Workload Scheduler: Administration Guide*.

Actions running against scheduling objects return empty tables

After running a monitor task, you run an action against the scheduling objects listed in the result table, but you get, as a result of the action, an empty table or window, and no error message is displayed. This occurs regardless of which action you try to run against the listed scheduling objects.

Cause and solution:

Check if the connection with the Tivoli Workload Scheduler engine where you run the task failed by doing the following:

1. In the **Configuration** window select Scheduler Connections.
2. Select in the list the engine used to run the browse task and click **Test Connection**

Note: The user ID you use to connect to the Dynamic Workload Console must belong either to the **TWSWEBUIAdministrator** or to the **TWSWEBUIConfigurator** groups to test the engine connection.

If the connection with the Tivoli Workload Scheduler engine is not active, ask the Tivoli Workload Scheduler administrator to restart the connection as described in the *IBM Tivoli Workload Scheduler Reference Guide*, and then rerun the action.

If the connection with the Tivoli Workload Scheduler engine is active, then, on that engine, check that:

- The Tivoli Workload Scheduler user running the command to list scheduling objects is authorized to do so. For more information about how to set user authorization, refer to the *IBM Tivoli Workload Scheduler Reference Guide*.
- The global property *enListSecChk* is set to enable on the Tivoli Workload Scheduler master domain manager. For more information about how to set global properties, refer to the *IBM Tivoli Workload Scheduler Planning and Installation Guide*.

Then rerun the action.

Default tasks are not converted into the language set in the browser

An existing user logs in to the Dynamic Workload Console using a browser where the language set is different from the language that was set in the browser the first time he logged in. In the Manage Tasks window, the default tasks are not translated into the new language.

Cause and solution:

The default tasks are created, using the current language set in the browser, when the new user logs into the Dynamic Workload Console for the first time. To have the default tasks translated into a different language, the WebSphere Application Server administrator must create a new Dynamic Workload Console user, and use that to login to the Dynamic Workload Console for the first time using a browser configured with the requested language. By doing this the default tasks are created using the requested language.

"Access Error" received when launching a task from the browser bookmark

A Dynamic Workload Console task has been saved in the list of bookmarks of the browser. You try to launch the task using the bookmark but you receive the following error message:

"User does not have access to view this page, use the browser back button to return to previous page."

Cause and solution:

You do not have the necessary role required to run the task. To run a task you must have a role that allows you to access the Dynamic Workload Console panels that are relevant to the type of task you need.

For more information about setting roles to work with the Dynamic Workload Console, see the Administration Guide, under the section about Configuring new users to access Dynamic Workload Console

If you close the browser window, processing threads continue in the background

You perform an action or make a selection and immediately close the browser window. You expect that processing terminated but the messages stored in the `SystemOut.log` file show that processing continued in the background.

Cause and solution:

This is normal behavior for any WEB application, when the client browser is closed no notification is delivered to the server according to the HTTP protocol specifications. This is the reason why the last triggered thread continues to process even after the browser window was closed. You do not need to run any action, just allow the thread to end.

The list of Available Groups is empty in the Enter Task Information window

You are creating a task, and you notice that in the Enter Task Information the list of Available Groups is empty. You are using an LDAP user registry.

Cause and solution:

Log into the Integrated Solutions Console as administrator and check the advanced LDAP configuration settings are correct as follows:

1. In the Navigation tree click **Security**.
2. Click **Secure administration, applications, and infrastructure**.
3. Check that the **Available realm definitions** field is set to **Standalone LDAP registry**.
4. Click **Configure**.
5. Click **Advanced Lightweight Directory Access Protocol (LDAP) user registry settings** under **Additional Properties**.
6. Verify that the settings for groups and users are correct for your configuration.

For more information about how to set these values, refer to: http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.express.doc/info/exp/ae/usec_advldap.html

JVM failure when working with the Dynamic Workload Console on a Red Hat Enterprise Linux (RHEL) Version 5 system

When working with the Dynamic Workload Console on a Red Hat Enterprise Linux Version 5 system, a user might see the error "Failed to find VM - aborting"

Cause and solution:

Red Hat Enterprise Linux Version 5 has a new security feature named 'Security Enhanced Linux', or SELinux for short. A weaker version of SELinux was included in Red Hat Enterprise Linux Version 4, and was disabled by default. Red Hat Enterprise Linux Version 5 defaults SELinux to enabled. SELinux helps to keep the host secure from certain types of malicious attacks. However, the default settings have been known in many cases to prevent Java from running properly.

To fix this issue, you can choose one of the following options:

- Configure SELinux so that it knows that the Dynamic Workload Console Java related processes are acceptable to run.
- Change the mode of SELinux to *Permissive* by entering `setenforce 0` on the command line. SELinux will be fully enabled again the next time the system is rebooted or if `setenforce 1` is entered on the command line.

Missing daylight saving notation in the time zone specification on Dynamic Workload Console 8.4 Fix Pack 1 and later

When using Dynamic Workload Console 8.4, the time zone is displayed using the *Daylight Saving*, or *Summer* notation, for example:

Europe/Paris (Central European Summer Time, GMT+1:00)

Starting from Dynamic Workload Console 8.4 Fix Pack 1, the *Summer* notation is no longer displayed and the time zone is expressed as follows:

Europe/Paris (Central European Time, GMT+1:00)

Cause and solution:

This is just a change in the standard time zone notation and does not affect the time conversion mechanisms. You can ignore this difference.

A "java.net.SocketTimeoutException" received

See the following scenario: "A "java.net.SocketTimeoutException" received" on page 166.

Blank page displayed (in High availability disaster recovery configuration)

You are in a high availability disaster recovery (HADR) configuration and the Dynamic Workload Console displays blank panels when trying to retrieve information from DB2.

Cause and solution:

When DB2 primary node stops, every Dynamic Workload Console request waits for a manual switch to a standby node.

If you have a HADR DB2 configuration related to your Tivoli Workload Scheduler engine, and you get an empty or blocked panel in the Dynamic Workload Console, verify that your primary node is up and running.

Some panels in Dynamic Workload Console might not be displayed correctly

Some panels in Dynamic Workload Console might not be displayed correctly.

Cause and solution:

This is due to problems in enabling the Java Authorization Contract for Containers (JACC)-based authorization.

To resolve this problem, run the **propagatePolicyToJACCProvider{-appNames appNames}** command in WebSphere Application Server:

On Windows systems

```
| "c:\Program Files\IBM\JazzSM\profile\bin\wsadmin.bat" -conntype SOAP
| -username "wasadmin" -password ***** -c "$AdminTask
| propagatePolicyToJACCProvider"
```

On UNIX systems

```
| /opt/IBM/JazzSM/profile/bin/wsadmin.sh -conntype SOAP -username
| wasadmin -password ***** -lang jython -c
| "AdminTask.propagatePolicyToJACCProvider()"
```

where

conntype

Specifies the type of connection to use.

username

Specifies a user name to be used by the connector to connect to the server if security is enabled in the server.

password

Specify a password to be used by the connector to connect to the server, if security is enabled in the server

lang

Specifies the language of the script file, the command, or an interactive shell.

c

Specifies to run a single command.

For more information, see WebSphere Application Server documentation.

Exceptions might not be displayed in Language-specific in the Dynamic Workload Console

When working with the Dynamic Workload Console, some exception might not be displayed correctly in the specific Language.

Cause and solution:

This might occur because if the master domain manager is installed in English language, all the exception that the master domain manager returns, are in English.

To solve the problem, you need to change the language of the machine where the engine is installed and restart master domain manager.

Extraneous exception logged in SystemOut

While working with the Dynamic Workload Console, the following exception message might be reported in the SystemOut.log file:

```
ConnException E   AWSJC0005E WebSphere Application Server
has given the following error:
CORBA NO_PERMISSION 0x0 No;
nested exception is:
  org.omg.CORBA.NO_PERMISSION:
>> SERVER (id=4773e3aa, host=axrsgpar0612.metlife.com) TRACE START:
>>   org.omg.CORBA.NO_PERMISSION: java.rmi.AccessException: ;
nested exception is:
  com.ibm.websphere.csi.CSIAccessException: SECJ0053E:
Authorization failed for
/UNAUTHENTICATED while invoking (Bean)ejb/com/ibm/tws/conn/engine
/ConnEngineHome getEngineInfo(com.ibm.tws.conn.util.Context):1 securityName:
/UNAUTHENTICATED;accessID: UNAUTHENTICATED is not granted any
of the required roles:
TWSAdmin   vmcid: 0x0   minor code: 0   completed: No
```

Cause and solution:

This exception message might be logged if you are using an engine for which the credentials are not stored. You can ignore the exception message. It is not an indication that the product is not functioning correctly.

Filtering task results might not work as expected

When you use the quick filtering feature to filter the list of results shown in Dynamic Workload Console tables, with engines version 9.1 or later, you must consider the following limitations:

Filtering on dates and duration

- You cannot filter by timezone and offset.
- You cannot filter using text strings in date columns.
- Even though the table of results shows dates in the following format: mm/dd/yyyy, the leading "0" is not considered when filtering. For example, 6/8 when filtering is considered as if it was 06/08.
- Even though the table of results shows duration time in the following format: hh:mm, the leading "0" is not considered when filtering. For example, 06:08 when filtering is considered as if it was 6:08. In durations like 00:01 the 00part cannot be matched with the search string 00, because all non significant digits are discarded. The proper way to search for that is with the string 0.

Filtering not supported

Quick filtering feature is not supported on the following columns:

- Information
- Node type

Filtering on job types

To filter for the following job types you must use the specified text:

Shadow Distributed

In the filter field, enter: **distributedShadowJob**

Shadow z/OS

In the filter field, enter: **zShadowJob**

Remote Command

In the filter field, enter: **remotecommand**

Database

In the filter field, enter: **database**

Executable

In the filter field, enter: **executable**

File Transfer

In the filter field, enter: **filetransfer**

IBM i In the filter field, enter: **ibmi**

J2EE In the filter field, enter: **j2ee**

Java In the filter field, enter: **java**

z/OS In the filter field, enter: **jcl**

MS SQL

In the filter field, enter: **mssqljob**

Provisioning

In the filter field, enter: **provisioning**

Web Services

In the filter field, enter: **ws**

Access Method

In the filter field, enter: **xajob**

OSLC Automation

In the filter field, enter: **oslcautomation**

Cause and solution:

This is due to a mismatch between how data is stored in the database and how it is shown from the Dynamic Workload Console.

You must set your user preferences so that dates are shown in short format (6/27/08 5:59) and you must use the specified strings to filter for job types.

Sorting task results might not work as expected

When you use the sorting feature to sort the list of results shown in Dynamic Workload Console tables, with engines version 9.1 or later, you must consider the following limitations:

- Sorting is not supported on Information column.
- When sorting on Node type column of Monitor workstations tasks, the sorting might result not correct.

Cause:

This is due to a mismatch between how data is stored in the database and how it is shown from the Dynamic Workload Console.

Chapter 11. Troubleshooting workload service assurance

Gives you troubleshooting information about workload service assurance by explaining how it works and how it exchanges information between the modules. In addition, it provides solutions to common problems.

This chapter provides information that is useful in identifying and resolving problems with the Workload Service Assurance feature. It includes the following sections:

- “Components involved in workload service assurance”
- “Exchange of information” on page 192
- “Common problems with workload service assurance” on page 192

Components involved in workload service assurance

Workload service assurance uses the following components to plan, monitor, and if necessary, intervene in the processing of jobs that are part of a critical network:

Planner

The planner component is triggered by the **JnextPlan** command. It includes a series of actions that result in the creation of the Symphony file on the master domain manager.

When workload service assurance is enabled, the planner calculates job streams and job networks, taking into consideration all "follows" dependencies in the new plan.

The planner then identifies all the jobs and job streams that are part of a critical network. These are jobs that are direct or indirect predecessors of a critical job. For each job, a critical start time is created and added to the Symphony file. It represents the latest time at which the job can start without putting the critical job deadline at risk. The plan information is then replicated in the database.

The Symphony file is subsequently distributed to all agents.

Plan monitor

The plan monitor component is introduced with the workload service assurance feature. It runs in the WebSphere Application Server on the master domain manager and is responsible for keeping track of the job streams and job network and for updating it when changes to the plan occur either because of the normal running of jobs or because of manual operations.

The plan monitor holds the information that is required to monitor the progress of the jobs involved in a critical network, for example critical start, planned start, estimated start, and risk level. It changes these values in response to changes in the plan, identified by the **batchman** process running on the master domain manager and communicated to the plan monitor using the `server.msg` file.

The information maintained by the plan monitor can be viewed on the Dynamic Workload Console in specialized views for critical jobs, allowing you easily to identify real and potential problems.

Agent processes (batchman and jobman)

Jobs in the critical network that are approaching the critical start time and have not started are promoted. The time at which the job is considered to be approaching its critical start time is determined by the global options setting `promotionOffset`.

The **batchman** process monitors the critical start time to determine if promotion is required and if so to schedule it at the highest job priority available in Tivoli Workload Scheduler. The **batchman** process also communicates with the **jobman** process, which is responsible for promoting the job at operating system level so that it receives more system resources when it starts. The operating system promotion is controlled by the local options settings `jm promoted nice` (UNIX) and `jm promoted priority` (Windows).

Exchange of information

Initially, the critical start time for jobs in the critical network is calculated by the planner and then recalculated, as required, by the plan monitor. Both of these components run on the master domain manager.

The critical start time is used by agents to determine when to promote a job. It is initially sent to the agent when the new Symphony file for the plan is distributed. Subsequent changes to critical start times are sent by the plan manager to agents using a Tivoli Workload Scheduler message. The agents update the local copy of the Symphony file.

The most common situations in which the plan monitor updates critical start times are:

- The Workload Designer functions on the Dynamic Workload Console or the **conman** command are used to modify jobs in the critical network. For example, predecessor jobs are added or cancelled.
- When **JnextPlan** is run to create the plan extension that includes the critical job, jobs in the original plan might be predecessors of the critical job and so be part of the critical network. In this case, critical start times are calculated by the plan monitor and sent in messages to the agents. This information is updated in both the local Symphony files and in the database where the plan data is replicated.

Common problems with workload service assurance

The following are problems that could occur when you are using Tivoli Workload Scheduler with workload service assurance enabled:

- “Critical start times not aligned”
- “Critical start times inconsistent” on page 193
- “Critical network timings change unexpectedly” on page 193
- “A high risk critical job has an empty hot list” on page 194

Critical start times not aligned

The values for critical start times in a critical network obtained from the appropriate `conman` commands on an agent are different from those displayed on the Tivoli Dynamic Workload Console.

Cause and solution:

Changes that affect the critical start times have been made to the plan since the Symphony file was sent to the agent. The changes are calculated on the master domain manager and sent to agents in messages. It is probable that the message has not reached the affected agent.

Check that the agent is active and linked to the master domain manager, either directly or by other domain managers.

Critical start times inconsistent

The values for critical start time in the chain of jobs in the critical network appears to be inconsistent. There are predecessor jobs that have critical start dates that are later than their successors.

Cause and solution:

This inconsistency occurs when critical start times are recalculated after some of the jobs in the critical network have completed. To optimize the calculation, new critical start times are only recalculated and updated for jobs that have not yet completed. The completed jobs retain the original critical start time. If a completed job is subsequently selected to be rerun, its critical start date will be recalculated.

Critical network timings change unexpectedly

Timings for jobs in the critical network change even though there have been no user actions related to the timing of jobs.

Cause and solution:

Changes can be made to timings because of a plan extension or because of the submission of jobs or job streams.

A critical job is consistently late

A job that is defined as critical is consistently late despite promotion mechanisms being applied to it and its predecessors.

Cause and solution:

Using the successful predecessors task, compare the planned start, the actual start, and the critical start of all the predecessors of the late job. Check if any of them have time values that are too close together or have a planned start time that is later than the critical start time.

In such a case, you can:

- Consider changing the timings of these jobs. For example, postpone the deadline if possible, or if the deadline must be maintained anticipate the start of some of the jobs.
- Consider redesigning your job streams to optimize the paths that are causing delays.
- Increase the value of the `promotionOffset` global option, so that jobs are promoted earlier.
- On the workstations where jobs are tending to be late, increase the `jm promoted nice` (UNIX) and `jm promoted priority` (Windows) local options, so that promoted jobs receive more system resources.

A high risk critical job has an empty hot list

A job that is defined as critical is shown to be at high risk, but its hot list is empty.

Cause and solution:

This normally only occurs if you have designed a critical job or a critical predecessor with a conflict which means it will always be late, for example a start restriction after the critical job deadline. The hot list is empty if either the job or job stream that is causing the problem doesn't have its follows dependencies resolved, or the job stream that is causing the problem is empty.

The only solution is to examine the critical path in detail and determine where the problem lies. The steps to resolving this problem are the same as those documented in "A critical job is consistently late" on page 193.

Chapter 12. Troubleshooting the fault-tolerant switch manager

Provides troubleshooting information about the fault-tolerant switch manager in terms of the event counter, the Ftbox, and link problems. It also provides solutions to some common problems with the backup domain manager.

This section describes how to address the potential problems related to the use of the fault-tolerant switch manager.

It is divided into the following sections:

- “Event counter”
- “Ftbox” on page 196
- “Troubleshooting link problems” on page 196
- “Common problems with the backup domain manager” on page 200

Event counter

The messages displayed in the log file concerning the event counter table are of three types:

- Messages that confirm the successful event counter initialization. No action is needed.
- Messages that the event counter reports related to problems not related to it. For example, they could reveal that the workstation received a message out of sequence. If action is required it does not impact the event counter.
- Messages that indicate that the event counter has failed. User action is needed to restore the counter.

This section concerns itself with this third type of messages.

Two processes can display this kind of error message:

Writer When an error message of this type is received from writer, the event counters stops. All messages received from the workstation which asked netman to activate writer, and from all its children, are ignored. This can lead to two situations:

- The workstation talking to writer is switched to a new manager. In this case the new manager asks for a counter table and receive a corrupt counter table. The replay protocol proceeds following the default behavior.
- Before the **switchmgr** operation can be performed, writer fails and is automatically restarted. In this case the counter mechanism partially repairs itself. New messages received by the process are stored in the counter, but the messages received by the writer from the moment the error message was displayed up to the point at which writer restarted are not tracked. The situation for a given workstation might be considered as reset only when the new instance of writer receives a message from it.

The situation is recovered after the next scheduled JnextPlan. If you need to recover more urgently, run **JnextPlan -for 0000** to refresh the Symphony file.

Mailman

When an error message of this type is received from mailman, the event counters stops. Mailman sets the IDs of all messages to 0. This means that there is a risk of duplication, because without the event counter, mailman is unable to properly sequence and process messages.

When the **switchmgr** is performed, and the new domain manager commences the replay protocol mechanism, for each message in the ftbox it looks at the position of the target workstation with respect to its own position in the tree:

- If the position of the target workstation in the workstation tree is higher than the new domain manager's (the workstation is either the domain manager or a full-status member of the parent domain of the domain where the **switchmgr** operation took place), the message is sent.
- If the position of the target workstation in the workstation tree is lower than the new domain manager's (the workstation either belongs to the domain where the **switchmgr** operation took place and it is not the new domain manager or is the domain manager or a full-status member of one of the child domains), the message is *not* sent.

The situation is recovered after JnextPlan.

Ftbox

If, on a full-status agent, you receive an error message concerning the ftbox, it means that the fault-tolerant backup domain manager feature is not working properly on that agent. Do not make this agent the new domain manager.

To restore the correct functionality of the feature on the instance, solve the problem as described in the error message, and restart the agent.

Troubleshooting link problems

When troubleshooting a link problem, the analysis is started from the master domain manager. The loss of the "F" flag at an agent indicates that some link had a problem. The absence of a secondary link can be located by matching the "W" flags found on the full-status fault-tolerant agent on the other side.

Consider the network shown in Figure 1 on page 197, where the workstation ACCT_FS, which is a full-status fault-tolerant agent, is not linked:

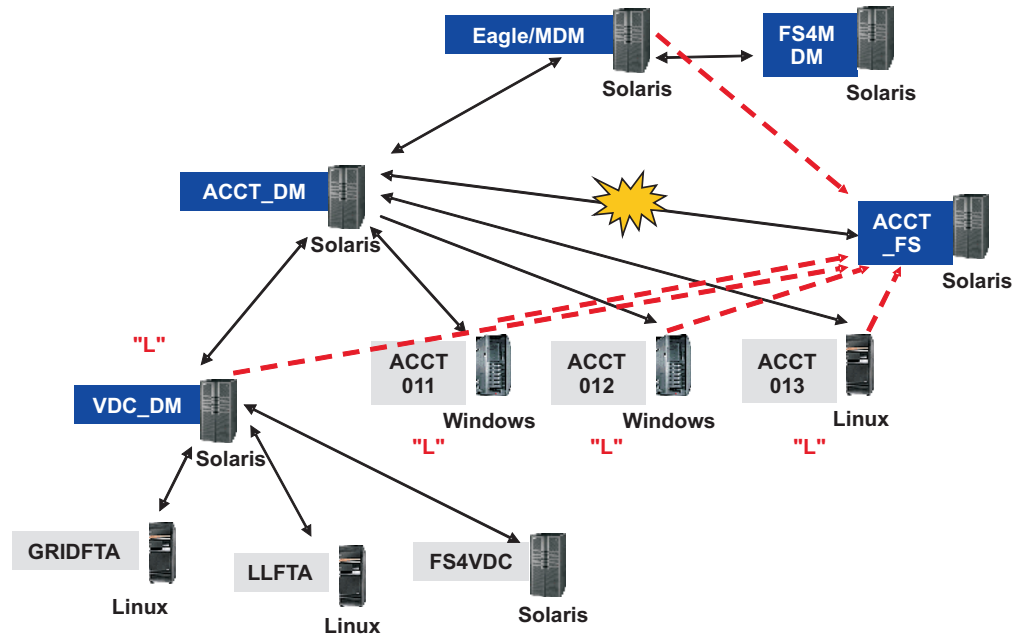


Figure 1. ACCT_FS has not linked

The key to Figure 1 is as follows (for those looking at this guide online or who have printed it on a color printer, the colors of the text and labels is indicated in parentheses, but if you are viewing it without the benefit of color, just ignore the color information):

White text on dark (blue) labels

CPUIDs of fault-tolerant agents in the master domain

Black text

Operating systems

Black text on grey labels

CPUIDs of standard agents in the master domain, or any agents in lower domains

Text (red) in "quotes"

Status of workstations obtained by running **conman sc @!@** at the master domain manager. Only statuses of workstations that return a status value are shown.

Black double-headed arrows

Primary links in master domain

Explosion

Broken primary link to ACCT_FS

Dotted lines (red)

Secondary links to ACCT_FS from the other workstations in the ACCT domain that could not be effected.

You might become aware of a network problem in a number of ways, but if you believe that a workstation is not linked, follow this procedure to troubleshoot the fault:

1. Use the command **conman sc @!@** on the master domain manager, and you can see that there is a problem with ACCT_FS, as shown in the example command output in Figure 2 on page 198:

```

$ conman sc @!@
Installed for user 'eagle'.
Locale LANG set to "C"
Schedule (Exp) 01/25/11 (#365) on EAGLE. Batchman LIVES. Limit: 20, Fence: 0,
Audit Level: 1
sc @!@
CPUID      RUN   NODE      LIMIT FENCE   DATE    TIME  STATE  METHOD  DOMAIN
EAGLE      365   *UNIX MASTER 20    0    01/25/11 05:59 I J      MASTERDM
FS4MDM     365   UNIX FTA    10    0    01/25/11 06:57 FTI JW      MASTERDM
ACCT_DM    365   UNIX MANAGER 10    0    01/25/11 05:42 LTI JW      DM4ACCT
ACCT011    365   WNT FTA     10    0    01/25/11 06:49 L I J      DM4ACCT
ACCT012    365   WNT FTA     10    0    01/25/11 06:50 L I J      DM4ACCT
ACCT013    365   UNIX FTA     10    0    01/25/11 05:32 L I J      DM4ACCT
ACCT_FS    363   UNIX FTA     10    0                      DM4ACCT
VDC_DM     365   UNIX MANAGER 10    0    01/25/11 06:40 L I J      DM4VDC
FS4VDC     365   UNIX FTA     10    0    01/25/11 06:55 F I J      DM4VDC
GRIDFTA    365   OTHR FTA     10    0    01/25/11 06:49 F I J      DM4VDC
GRIDXA     365   OTHR X-AGENT 10    0    01/25/11 06:49 L I J      gridage+ DM4VDC
LLFTA      365   OTHR FTA     10    0    01/25/11 07:49 F I J      DM4VDC
LLXA       365   OTHR X-AGENT 10    0    01/25/11 07:49 L I J      llagent  DM4VDC
$

```

Figure 2. Example output for **conman sc @!@** run on the master domain manager

- From the ACCT_DM workstation run **conman sc**. In this case you see that all the writer processes are running, except for ACCT_FS. These are the primary links, shown by the solid lines in Figure 1 on page 197. The output of the command in this example is as shown in Figure 3:

```

$ conman sc
TWS for UNIX (SOLARIS)/CONMAN 8.6 (1.36.2.21)
Licensed Materials Property of IBM
5698-WKB
(C) Copyright IBM Corp 1998,2011
US Government User Restricted Rights
Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Installed for user 'dm010'.
Locale LANG set to "C"
Schedule (Exp) 01/25/11 (#365) on ACCT_DM. Batchman LIVES. Limit: 10, Fence: 0,
Audit Level: 1
sc
CPUID      RUN   NODE      LIMIT FENCE   DATE    TIME  STATE  METHOD  DOMAIN
EAGLE      365   UNIX MASTER 20    0    01/25/11 05:59 LTI JW      MASTERDM
ACCT_DM    365   *UNIX MANAGER 10    0    01/25/11 05:42 I J      DM4ACCT
ACCT011    365   WNT FTA     10    0    01/25/11 06:49 LTI JW      DM4ACCT
ACCT012    365   WNT FTA     10    0    01/25/11 06:50 LTI JW      DM4ACCT
ACCT013    365   UNIX FTA     10    0    01/25/11 05:32 LTI JW      DM4ACCT
ACCT_FS    363   UNIX FTA     10    0                      DM4ACCT
VDC_DM     365   UNIX MANAGER 10    0    01/25/11 06:40 LTI JW      DM4VDC
$

```

Figure 3. Example output for **conman sc** run on the domain manager

- From the ACCT_FS workstation run **conman sc**. In this case you see that there are no writer processes running. These are the secondary links, shown with the

dashed lines in Figure 1 on page 197. The output of the command in this example is as shown in Figure 4:

```
$ conman sc
Installed for user 'dm82'.
Locale LANG set to "C"
Schedule (Exp) 01/24/11 (#364) on ACCT_FS. Batchman LIVES. Limit: 10, Fence: 0
, Audit Level: 1
sc @!@
CPUID      RUN   NODE      LIMIT FENCE   DATE   TIME   STATE  METHOD  DOMAIN
EAGLE      363   UNIX MASTER  20    0
FS4MDM     363   UNIX FTA    10    0
ACCT_DM    363   UNIX MANAGER 10    0
ACCT011    363   WNT FTA    10    0
ACCT012    363   WNT FTA    10    0
ACCT013    363   UNIX FTA    10    0
ACCT_FS    363   *UNIX FTA    10    0
VDC_DM     363   UNIX MANAGER 10    0
FS4VDC     363   UNIX FTA    10    0
GRIDFTA    363   OTHR FTA    10    0
GRIDXA     363   OTHR X-AGENT 10    0
$
```

Figure 4. Example output for **conman sc** run on the unlinked workstation

4. If a network problem is preventing ACCT_FS from linking, resolve the problem.
5. Wait for ACCT_FS to link.
6. From the ACCT_FS workstation, run **conman sc @!@**. If the workstation has started to link, you can see that a writer process is running on many of the workstations indicated in Figure 1 on page 197. Their secondary links have now been made to ACCT_FS. The workstations that have linked have an "F" instead of their previous setting. This view also shows that the master domain manager has started a writer process running on ACCT_FS. The output of the command in this example is as shown in Figure 5 on page 200:

```

$ conman sc @!@
Installed for user 'dm82'.
Locale LANG set to "C"
Schedule (Exp) 01/24/11 (#364) on ACCT_FS. Batchman LIVES. Limit: 10, Fence: 0
, Audit Level: 1
sc @!@
CPUID      RUN   NODE      LIMIT FENCE   DATE    TIME    STATE  METHOD  DOMAIN
EAGLE      371   UNIX MASTER 20    0    01/25/11 10:16 F I JW   MASTERDM
FS4MDM     370   UNIX FTA    10    0                               MASTERDM
ACCT_DM    371   UNIX MANAGER 10    0    01/25/11 10:03 LTI JW   DM4ACCT
ACCT011    369   WNT FTA     10    0                               DM4ACCT
ACCT012    371   WNT FTA     10    0    01/25/11 11:03 F I JW   DM4ACCT
ACCT013    371   UNIX FTA     10    0    01/25/11 09:54 F I JW   DM4ACCT
ACCT_FS    371   *UNIX FTA    10    0    01/25/11 11:08 F I J    DM4ACCT
VDC_DM     371   UNIX MANAGER 10    0    01/25/11 10:52 F I JW   DM4VDC
FS4VDC     371   UNIX FTA     10    0    01/25/11 11:07 F I J    DM4VDC
GRIDFTA    371   OTHR FTA     10    0    01/25/11 11:01 F I J    DM4VDC
GRIDXA     371   OTHR X-AGENT 10    0    01/25/11 11:01 L I J    gridage+ DM4VDC
LLFTA      371   OTHR FTA     10    0    01/25/11 12:02 F I J    DM4VDC
LLXA       371   OTHR X-AGENT 10    0    01/25/11 12:02 L I J    llagent  DM4VDC
$

```

Figure 5. Example output for `conman sc @!@` run on the unlinked workstation

- Another way of checking which writer processes are running on ACCT_FS is to run the command: `ps -ef | grep writer` (use Task Manager on Windows). The output of the `ps` command in this example is as shown in Figure 6:

```

$ ps -ef | grep writer
dm82 1363 616 0 06:43:11 ? 0:01 /usr/local/Tivoli/dm82/bin/write -- 2001 EAGLE MAILMAN UNIX 8.6 9
dm82 1317 616 0 06:42:21 ? 0:01 /usr/local/Tivoli/dm82/bin/write -- 2001 ACCT_DM MAILMAN UNIX 8.6 9
dm82 1337 616 0 06:42:25 ? 0:01 /usr/local/Tivoli/dm82/bin/write -- 2001 ACCT013 MAILMAN UNIX 8.6 9
dm82 1338 616 0 06:42:27 ? 0:01 /usr/local/Tivoli/dm82/bin/write -- 2001 VDC_DM MAILMAN UNIX 8.6 9
dm82 1364 616 0 06:51:48 ? 0:01 /usr/local/Tivoli/dm82/bin/write -- 2001 ACCT012 MAILMAN WNT 8.6 9
dm82 1336 616 0 06:42:24 ? 0:00 /usr/local/Tivoli/dm82/bin/write -- 2001 ACCT011 MAILMAN WNT 8.6 9
$

```

Figure 6. Example output for `ps -ef | grep writer` run on the unlinked workstation

- To determine if a workstation is fully linked, use the **Monitor Workstations** list in the Dynamic Workload Console.

Common problems with the backup domain manager

The following problems could be encountered with the fault-tolerant backup domain manager (note that a backup domain manager is an agent with the *full status* attribute set):

- “The Symphony file on the backup domain manager is corrupted.” on page 201
- “Processes seem not to have been killed on previous UNIX domain manager after running `switchmgr`” on page 201
- “In a scenario involving more than one `switchmgr` command, agent cannot relink” on page 201

The Symphony file on the backup domain manager is corrupted.

When switching to the backup domain manager from the master domain manager, the Symphony file on the backup domain manager might become corrupted.

Cause and solution:

The "thiscpu" variable in the localopts file does not match the workstation name. Change the variable to match the workstation name and the problem no longer occurs.

Processes seem not to have been killed on previous UNIX domain manager after running switchmgr

You want to use the switch manager facility. You first stop all Tivoli Workload Scheduler processes on the domain manager and then you run **switchmgr**, which completes successfully. However, after running **%sc @!@**, the J flag state is given for the domain manager where you stopped the processes.

Cause and solution:

When a **shutdown** command is sent to a workstation, some unexpected output might be shown by the status of the processes shown by conman, as follows:

- The J flag relative to the shut workstation remains active (no message indicating that jobman is not running can be transmitted because mailman is also not running).
- Conman output on the shutdown workstation is not up-to-date (the Symphony file is not updated on the shutdown workstation)
- The shutdown workstation seems linked from its father and son workstations (no unlink operation is run by the writers on the workstation that is shutting down)
- Both F or L flags might be displayed, depending on the messages processed by mailman before unlinking and stopping.

The correct link situation is restored as soon as a new link attempt is made to the workstation, either manually, or automatically (after 10 minutes).

The **shutdown** command must be sent only in critical situations (where a workstation is shutting down, for example).

To avoid these problems, precede the **shutdown** command with an **unlink @!@** or **stop** command.

In a scenario involving more than one switchmgr command, agent cannot relink

You have been using the **switchmgr** command to switch to backup master domain manager, and then back to the master domain manager, but an agent might not have relinked to the original master domain manager.

Cause and solution:

The complex interaction of variables, environments, network conditions, and linking and relinking events can sometimes prevent an agent from relinking correctly.

No events or messages are lost, you can repeat the use of **switchmgr**, if necessary, and the performance of the network is not normally impacted because one agent is out of communication.

If only one agent is involved the easiest solution is to manually relink it.

However, to avoid having to identify and specifically relink the non-linked agent or agents, you can, in any case, issue the following command, which automatically relinks all agents without needing to specifically identify the unlinked ones:

JnextPlan -for 0000

Chapter 13. Synchronizing the database with the Symphony file

If you suspect the plan data loaded in the database is not up-to-date, you can run **planman resync** to update the database with the latest information in the Symphony file.

Note: If the message box, mirrorbox.msg, responsible for synchronizing the database with the Symphony file becomes full, for example, the database is unavailable for a long period of time, then a **planman resync** is automatically issued so that the plan is fully reloaded in the database.

The procedure requires you to run the following step on the master domain manager. If you run the command on the backup master domain manager when it is not acting as the master domain manager, the plan data is not replicated in the database.

On the master domain manager, issue the following command:
planman resync

All of the plan data currently stored in the Symphony file is replicated in database tables in the database.

For the complete command-line syntax for the **planman** command, refer to the *Tivoli Workload Scheduler: User's Guide and Reference*.

To see the database views containing information about the objects in the plan, see the views beginning with "PLAN_" in *Tivoli Workload Scheduler: Database Views*.

Chapter 14. Corrupt Symphony file recovery

Explains the symptoms of Symphony file corruption and links you to tasks that can recover the file on the master domain manager, fault-tolerant agent, or lower domain manager.

This section describes how to diagnose and fix a corruption of the Symphony file.

Symphony file corruption is a rare event, and a potential corruption must be verified before taking action. Common symptoms are the following:

- A specific message informing you that the Symphony file is corrupt
- A shutdown of various processes (especially batchman) with error messages referring to problems with the Symphony file in the stdlist

The normal reason for the corruption of the Symphony file is a full file system. This can be avoided by regular monitoring of the file system where Tivoli Workload Scheduler is installed.

The procedure is different, depending on the location of the corrupt Symphony file.

Recovery procedure on a master domain manager

If a **Symphony** file is corrupt on a master domain manager, there are several ways in which you can regenerate it.

The **Symphony** file can be regenerated in the following ways:

1. Using the backup master domain manager
2. Using the logman and ResetPlan commands
3. Using the latest archived plan

Recovering using the backup master domain manager

If the Symphony file is corrupt on a master domain manager, it can be regenerated using the backup master domain manager.

The regeneration of the Symphony file causes some minor loss of data. The following procedure indicates what is lost.

The prerequisite for the procedure is to have a backup master domain manager already available. A backup master domain manager is a fault-tolerant agent in the master domain with its `fullstatus` attribute set to *yes*.

Note: If you have not already created a backup master domain manager, the Symphony file cannot be recovered and the processing it contains is lost.

The procedure requires you to take the following steps on either the master domain manager or the backup master domain manager:

Note: The steps must be followed in strict order; each step description below is prefaced by the identification of the workstation on which it must be performed.

1. On the backup master domain manager, do the following:

- a. Issue the **switchmgr** command.
 - b. Verify that the backup master domain manager is acting as the master domain manager.
2. From the new master domain manager set the job "limit" on the old master domain manager to "0", using **conman** or the Dynamic Workload Console. This prevents jobs from launching.
3. On the original master domain manager do the following:
 - a. Shut down all Tivoli Workload Scheduler processes
 - b. Rename the Sinfonia file and the corrupt Symphony file (any names will do).
4. On the current master domain manager (previous backup master domain manager) do the following:
 - a. Verify that it is linked to all agents *except* the old master domain manager.
 - b. Shut down all Tivoli Workload Scheduler processes (unlink from all agents).
 - c. Rename Sinfonia as Sinfonia.orig
 - d. Copy Symphony to Sinfonia

You now have identical Symphony and Sinfonia files.
5. On the original master domain manager do the following:
 - a. Issue a **StartUp** from the operating system's command line, to start the **netman** process.
 - b. Verify that the process remains active.
6. On the current master domain manager (previous backup master domain manager) do the following:
 - a. Issue a **StartUp** from the operating system's command line, to start the **netman** process.
 - b. Issue a **conman start**, or use the Dynamic Workload Console to start the current master domain manager.
 - c. Issue a link to the original master domain manager.

This action sends the Symphony file to the original master domain manager.
7. On the original master domain manager do the following:
 - a. Verify that the Symphony file is present and is the correct size (same as on the current master domain manager (previous backup master domain manager))
 - b. Verify that all Tivoli Workload Scheduler processes are active.
8. On the current master domain manager (previous backup master domain manager) verify that the original master domain manager is linked.
9. On the original master domain manager do the following:
 - a. Set the job "limit" on the old master domain manager to the previous level, using **conman** or the Dynamic Workload Console.

Jobs can commence launching.

 - b. Verify that the original master domain manager has the current job status for all agents.
 - c. Issue the **switchmgr** command to switch control back to the original master domain manager.

Following this procedure some information is lost, in particular, any events that were suspended on the master domain manager when you started the recovery procedure.

If this procedure cannot be performed, try using the procedure that uses the **logman** and **ResetPlan** commands: "Recover using the logman and ResetPlan commands."

Recover using the logman and ResetPlan commands

The following procedures can also be used to recover a corrupt Symphony file on the master domain manager.

These procedures do not recover as much data as "Recovering using the backup master domain manager" on page 205, but they might be useful if that procedure cannot be performed.

The procedure that makes use of **ResetPlan** might result in a more complete recovery, but it is more demanding in time since it scratches both the production and the preproduction plans. The preproduction plan will be created again based on the modeling information stored in the database when you later generate a new production plan. This means that the new production plan will contain all job stream instances scheduled to run in the time frame covered by the plan regardless of whether or not they were already in COMPLETE state when the plan was scratched.

You should first run the recovery procedure that makes use of **logman**. If you do not obtain satisfactory results, run the other one.

Neither procedure requires the use of a backup master domain manager.

Recovering the Symphony file using the logman command

Describes how to recover from a corrupt Symphony file using the **logman** command.

Perform these steps on the master domain manager:

1. Set the job "limit" to "0" on all the workstations by using **conman**, the Dynamic Workload Console, or the Job Scheduling Console. If you are using **conman**, run the following command:

```
conman "limit cpu=@!@;0;noask"
```

This command prevents jobs from launching.
2. Shut down all Tivoli Workload Scheduler processes on the master domain manager.
3. Run **logman -prod** to update the preproduction plan with the information for the job streams that are in COMPLETE state.
4. Run **planman showinfo** and check for the first incomplete job stream instance.
5. Run **ResetPlan**.
6. Run **JnextPlan**, setting the **-from** parameter to the start time of the first incomplete job stream instance in the preproduction plan (acquired from the output of **planman showinfo**) and the **-to** parameter to the end date of your plan (or to the following day). Only incomplete Job Scheduler instances will be included in the new Symphony file. If the instance of the first incomplete job stream is very old the new plan creation can take a long time. The incomplete jobs and job streams that are recreated with the **JnextPlan -from** parameter are those present in the database when the command is run.
7. Check the created plan and verify that all the jobs and job streams in the plan have the correct status.

8. Ensure that you want to run all the instances in the plan, deleting those that you do not want to run.
9. All the submitted job streams are not carried forward. Resubmit them.
10. Reset the job "limit" to its previous value. The Symphony file is distributed and production starts again.

Note: The status of the jobs and the job streams after you run the recovery procedure is reset to HOLD or READY.

Recovering with the use of the ResetPlan command

Follow these steps on the master domain manager :

1. Set the job "limit" to "0", using **conman** or the Dynamic Workload Console. This prevents jobs from launching.
2. Shut down all Tivoli Workload Scheduler processes on the master domain manager.
3. Run **ResetPlan -scratch**.
4. Run **JnextPlan**, setting the **-from** and **-to** parameters to cover the period for which there are still outstanding jobs.
5. Check the created plan and ensure that you want to run all the instances it contains, deleting those that you do not want to run.
6. Reset the job "limit" to the previous value. The Symphony file is distributed and production recommences.

Recovering the plan from the latest archived plan

You can recover a corrupted plan on a master domain manager using the latest archived plan, however; this is possible only if you have performed some configuration steps prior to the occurrence of the file corruption.

The following procedure recovers a corrupted plan using **ResetPlan**. The plan is recovered using the latest archived plan and any events logged throughout the day are written to a new event message file. The last archived Symphony file is copied into the current Symphony file and then **JnextPlan** is run to apply the events from the evtlog.msg file.

Restriction: Before you can perform the recovery procedure, you must have completed some configuration steps prior to the file corruption occurrence.

1. **Complete the following configuration steps so that you can use the recovery procedure in the future if it becomes necessary:**
 - a. In the localopts file, add the following attribute and value: **bm log events = ON**.
 - b. Optionally, customize the path where Tivoli Workload Scheduler creates the evtlog.msg event file by setting the **bm log events path** property in the localopts file. If you do not modify this setting, the evtlog.msg event file is created in the following default location: **<TWA_INST_DIR>/TWS**.
 - c. Stop and start all Tivoli Workload Scheduler processes or run **JnextPlan** to create the evtlog.msg file.
 - d. If necessary, you can configure the maximum size of both the evtlog.msg and Intercom.msg event files as follows:

```
evtsize -c evtlog.msg 500000000
evtsize -c Intercom.msg 550000000
```


Note: The default size of these event files is 10 MB. When the maximum size is reached, events are no longer logged to these files and the recovery procedure is unable to recover them and any that follow. Moreover, the following **BATCHMAN** warning is logged to <TWA_INST_DIR>/TWS/stdlist/traces/YYYYMMDD_TWSMERGE.log:

```
13:11:51 18.10.2012|BATCHMAN:+ WARNING:Error writing in evtlog:
    AWSDEC003I End of file on events file.
13:11:51 18.10.2012|BATCHMAN:*
13:11:51 18.10.2012|BATCHMAN:* AWSBHT160E The EvtLog message file is full,
    events will not be logged until a new Symphony is produced. Recovery with
    event reapply is no more possible until that time.
13:11:51 18.10.2012|BATCHMAN:*
```

If you encounter this problem increase the size of the evtlog.msg and Intercom.msg event files.

Consider that for 80,000 jobs and a Symphony file of size 40 MB, the evtlog.msg file is approximately 70 MB in size.

Important: The Intercom.msg maximum size should always be set to a value greater than the maximum size of evtlog.msg

In the <TWA_INST_DIR>/TWS/stdlist/traces/YYYYMMDD_TWSMERGE.log trace file, the **BATCHMAN** process logs an informational line containing the expected size of the evtlog.msg queue. For example:

```
19:02:06 14.10.2012|BATCHMAN:INFO:0.25 MB of events to log during this
    batchman run
```

If Intercom.msg reaches the maximum size during the recovery procedure, **batchman** stops.

- e. If the file system where evtlog.msg resides runs out of space, a **BATCHMAN** warning is logged to <TWA_INST_DIR>/TWS/stdlist/traces/YYYYMMDD_TWSMERGE.log as follows:

```
13:10:36 16.10.2012|BATCHMAN:+ WARNING:Error writing in evtlog:
    AWSDEC002E An internal error has occurred.
The following UNIX system error occurred on an events file:
    "No space left on device" at line = 3517.
```

2. Complete the recovery procedure:

- a. Ensure the Tivoli Workload Scheduler processes are stopped. Run conman stop to stop them.
- b. Copy the information retrieved by running the **planman showinfo** command.
- c. Run ResetPlan. The corrupted Symphony file is archived in the schedlog folder.
- d. Copy the second last Symphony file archived in the schedlog folder, and not the most recent one which is the corrupted file. For example, on UNIX, submit the following command:


```
cp -p /opt/ibm/TWA/TWS/schedlog/MYYYYMMDDhhmm /opt/ibm/TWA/TWS/Symphony
```
- e. Run **JnextPlan** as follows using the information retrieved when you ran **planman showinfo**:

```
JnextPlan -from MM/DD/YYYY hhmm TZ Timezone -for hhhmm
```

where,

-from Production plan start time of last extension.

-for Production plan time extension.

When running this procedure, consider that the run number, that is, the total number of times the plan was generated, is automatically increased by one.

Job stream instances that have already completed successfully at the time this procedure is run are not included in the recovered plan.

After you have performed the recovery procedure, the workstation limit is set to 0 and the evtlog.msg queue is cleared with each successive run of **JnextPlan**.

Recovery procedure on a fault-tolerant agent or lower domain manager

If the Symphony file is corrupt on a lower level domain manager, or on a fault-tolerant agent, it can be replaced.

Complete removal and replacement of the Symphony file causes some loss of data. The following procedure minimizes that loss and indicates what is lost.

The procedure involves two agents, the agent where the Symphony file is corrupt and its domain manager.

Note: Where the agent is a top level domain manager (below the master), or a fault-tolerant agent in the master domain, the manager is the master domain manager.

The procedure is as follows:

1. On the domain manager, unlink the agent which is having the Symphony file problem.
2. On the agent do the following:
 - a. Stop the agent if it has not yet failed. You do not need to shut it down.
 - b. Delete the Symphony and the Sinfonia files from the agent workstation. Alternatively you can move them to a different location on the agent workstation, or rename them.
3. On the domain manager do the following:
 - a. Back up the Sinfonia file if you want to be able to restore the original situation after completion. This is not an obligatory step, and no problems have been reported from not performing it.
 - b. Ensure that no agent is linking with the domain manager, optionally stopping the domain manager agent.
 - c. Copy the domain manager's Symphony file to the Sinfonia file, replacing the existing version.
 - d. Restart the domain manager agent if necessary.
 - e. Link the agent and wait for the Symphony file to copy from the domain manager to the agent. The agent automatically starts.
 - f. Optionally restore the Sinfonia file from the backup you took in step 3a. This restores the original situation, but with the agent now having an uncorrupted Symphony file. This is not an obligatory step, and no problems have been reported from not performing it.

Following this procedure some information is lost, in particular, the contents of the Mailbox.msg message and the tomaster.msg message queues. If state information about a job was contained in those queues, such that the Symphony file on the

domain manager was not updated by the time the Sinfonia file is replaced (step 3c on page 210), that job is rerun. To avoid that event, add these steps to the procedure immediately before step 3a on page 210:

1. Make a list of jobs that ran recently on the agent.
2. At the domain manager, change their states to either SUCC or ABEND, or even cancel them on the domain manager.

Note: if you set the states of jobs to SUCC, or cancel them, any successor jobs would be triggered to start. Ensure that this is the acceptable before performing this action.

This way these jobs are not rerun.

Recovery procedure on a fault-tolerant agent with the use of the `resetFTA` command

If the Symphony file is corrupt on a fault-tolerant agent, you can use the `resetFTA` command to automate the recovery procedure.

Complete removal and replacement of the Symphony file causes some loss of data, for example events on job status, or the contents of the Mailbox.msg message and the tomaster.msg message queues. If state information about a job was contained in those queues, that job is rerun. The following procedure minimizes that loss and indicates what is lost. It is recommended that you apply this procedure with caution.

The procedure renames the Symphony, Sinfonia, *.msg files on the fault-tolerant agent where the Symphony corruption occurred and generates an updated Sinfonia file, which is sent to the fault-tolerant agent. You can therefore resume operations quickly on the affected fault-tolerant agent, minimize loss of job and job stream information, and reduce recovery time.

The procedure involves two agents, the fault-tolerant agent where the Symphony file is corrupt and its domain manager.

You can start the command from any Tivoli Workload Scheduler workstation, with the exception of the fault-tolerant agent where the corruption occurred. Connection to the target fault-tolerant agent and to its domain manager is established using the `netman` port number. The default port number is 31111.

When you start the `resetFTA` command, the following operations are performed in the specified order:

on the fault-tolerant agent

- The following files are renamed:
 - Appserverbox.msg
 - clbox.msg
 - Courier.msg
 - Intercom.msg
 - Mailbox.msg
 - Monbox.msg
 - Moncmd.msg
 - Symphony

- Sinfonia

The operations are performed asynchronously, to ensure that all target files have been renamed before starting the procedure on the domain manager.

on the domain manager

1. A backup of the Sinfonia file is created.
2. The Symphony file is copied to the Sinfonia file.
3. The target fault-tolerant agent is linked.
4. The updated Sinfonia file is sent to the target fault-tolerant agent.

The syntax of the command is as follows:

Syntax

resetFTA `cpu`

Arguments

cpu Is the fault-tolerant agent to be reset.

This command is not available in the Dynamic Workload Console.

For more information, see the section about the **resetfta** command in *Tivoli Workload Scheduler: User's Guide and Reference*.

Appendix A. Support information

If you have a problem with your IBM software, you want to resolve it quickly. This section describes the following options for obtaining support for IBM software products:

- “IBM Support Assistant”
- “Searching knowledge bases” on page 214
- “Obtaining fixes” on page 215
- “Receiving support updates” on page 215
- “Contacting IBM Software Support” on page 216

IBM Support Assistant

The IBM Support Assistant is a free, stand-alone application that you can install on any workstation. You can then enhance the application by installing product-specific add-on modules for the IBM products you use.

The IBM Support Assistant saves you time searching product, support, and educational resources. The IBM Support Assistant helps you gather support information when you need to open a problem management record (PMR), which you can then use to track the problem.

The product-specific add-on modules provide you with the following resources:

- Support links
- Education links
- Ability to submit problem management reports

The IBM Support Assistant website is at <http://www.ibm.com/software/support/isa/>. Use this site for the following:

- Obtain general information about the IBM Support Assistant
- Choose the IBM Support Assistant more appropriate for your needs, and perform the following actions:

Download the IBM Support Assistant Lite for your product

To quickly collect diagnostic files to solve problems faster. This is a special offering of the IBM Support Assistant that contains only the data collection component customized for a specific product. IBM Support Assistant Lite provides quick deployment of IBM Support Assistant's data collection tool. It is customized to automate product-specific data collection. You can run ISA Lite to do data collection for your product without ever installing ISA or your ISA add-on.

Download and install the IBM Support Assistant Workbench and the add-on for Tivoli Workload Scheduler

To benefit from concurrent search, media viewer, guided troubleshooter, diagnostic tools, data collectors, service request submission, and other features. IBM Support Assistant can be customized for over 350 products.

Note: To locate and download the add-on for a product, use the IBM Support Assistant's interface. Full instructions about how to use the application and add-on are provided within the interface. Add-ons are

available at release level at the following URL: <http://www.ibm.com/support/docview.wss?&uid=swg27013117>).

If you cannot find the solution to your problem in the IBM Support Assistant, see "Searching knowledge bases."

Searching knowledge bases

You can search the available knowledge bases to determine if your problem was already encountered and is already documented.

Search the local information center

IBM provides extensive documentation that you can install on your local computer or on an intranet server. You can use the search function of this information center to query conceptual information, instructions for completing tasks, and reference information.

The information center can be found online at http://publib.boulder.ibm.com/infocenter/tivihelp/v47r1/index.jsp?topic=/com.ibm.tivoli.itws.doc_8.6/welcome_TWA.html, from where you can download and install the information center locally, or on your intranet server.

Search the Internet


If you cannot find an answer to your question in the information center, search the Internet for the latest, most complete information that might help you resolve your problem.

To search multiple Internet resources for your product, use the **Web search** topic in your information center. In the navigation frame, click **Troubleshooting and support** → **Searching knowledge bases** and select **Web search**. From this topic, you can search a variety of resources, including the following:

- IBM technical notes (Technotes)
- IBM downloads
- IBM Redbooks®
- IBM developerWorks®
- Forums and newsgroups
- Google


Search the IBM support website

The IBM software support website has many publications available online, one or more of which might provide the information you require:

1. Go to the IBM Software Support website (<http://www.ibm.com/software/support>).
2. Select **Tivoli** under the **Select a brand and/or product** heading.
3. Select **IBM Tivoli Workload Scheduler** under **Select a product**, and click the "Go" icon: . The Tivoli Workload Scheduler support page is displayed.
4. In the **IBM Tivoli Workload Scheduler support** pane click **Documentation**, and the documentation page is displayed.
5. Either search for information you require, or choose from the list of different types of product support publications in the **Additional Documentation support links** pane:
 - Information center

- Manuals
- IBM Redbooks
- White papers



If you click on **Information center** the Tivoli Workload Scheduler Information Center page opens, otherwise a search for the selected documentation type is performed, and the results displayed.

6. Use the on-screen navigation to look through the displayed list for the document you require, or use the options in the **Search within results for** section to narrow the search criteria. You can add **Additional search terms** or select a specific **Document type**. You can also change the sort order of the results (**Sort results by**). Then click the search icon to start the search: .

To access some of the publications you need to register (indicated by a key icon beside the publication title). To register, select the publication you want to look at, and when asked to sign in follow the links to register yourself. There is also a FAQ available on the advantages of registering.

Obtaining fixes

A product fix might be available to resolve your problem. To determine what fixes are available for your IBM software product, follow these steps:

1. Go to the IBM Software Support website (<http://www.ibm.com/software/support>).
2. Select **Tivoli** under the **Select a brand and/or product** heading.
3. Select **IBM Tivoli Workload Scheduler** under **Select a product** and click the "Go" icon: . The Tivoli Workload Scheduler support page is displayed.
4. In the **IBM Tivoli Workload Scheduler support** pane click **Download**, and the download page is displayed.
5. Either choose one of the displayed most-popular downloads, or click **View all download items**. A search for the downloads is performed, and the results displayed.
6. Use the on-screen navigation to look through the displayed list for the download you require, or use the options in the **Search within results for** section to narrow the search criteria. You can add **Additional search terms**, or select a specific **Download type**, **Platform/Operating system**, and **Versions**. Then click the search icon to start the search: .
7. Click the name of a fix to read the description of the fix and to optionally download the fix.

For more information about the types of fixes that are available, see the *IBM Software Support Handbook* at <http://www14.software.ibm.com/webapp/set2/sas/f/handbook/home.html>.

Receiving support updates

To receive email notifications about fixes and other software support news, follow these steps:

1. Go to the IBM Software Support website at <http://www.ibm.com/software/support>.
2. Click **My notifications** under the **Stay informed** heading in the upper-right corner of the page.

3. If you have already registered for **My support**, sign in and skip to the next step. If you have not registered, click **register now**. Complete the registration form using your email address as your IBM ID and click **Submit**.
4. Follow the instructions on the page for subscribing to the information you require, at the frequency you require, for the products you require.

If you experience problems with the **My notifications** feature, you can obtain help in one of the following ways:

Online

Send an email message to erchelp@ca.ibm.com, describing your problem.

By phone

Call 1-800-IBM-4You (1-888 426 4409).

Contacting IBM Software Support

IBM Software Support provides assistance with product defects.

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli, Lotus®, and Rational® products, as well as DB2 and WebSphere products that run on Windows, or UNIX operating systems), enroll in Passport Advantage® in one of the following ways:

Online

Go to the Passport Advantage website at http://www.lotus.com/services/passport.nsf/WebDocs/Passport_Advantage_Home and click **How to Enroll**.

By phone

For the phone number to call in your country, go to the IBM Software Support website support handbook contacts page at <http://www14.software.ibm.com/webapp/set2/sas/f/handbook/contacts.html>, and click **IBM Directory of worldwide contacts** or select your geographical area for a list of contacts.

- For customers with IBMLink, CATIA, Linux, S/390®, System i®, System p®, System z®, and other support agreements, go to the IBM Support Line website at <http://www.ibm.com/services/us/index.wss/so/its/a1000030/dt006>.
- For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in System i, System p, and System z environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage website at <http://www.ibm.com/servers/eserver/techsupport.html>.

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States. From other countries, go to the contacts page of the *IBM Software Support Handbook on the Web* at <http://www14.software.ibm.com/webapp/set2/sas/f/handbook/contacts.html> and click the name of your geographic region for phone numbers of people who provide support for your location.

To contact IBM Software support, follow these steps:

1. "Determine the business impact"
2. "Describe problems and gather information"
3. "Submit problems"

Determine the business impact

When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem that you are reporting. Use the following criteria:

Severity 1

The problem has a *critical* business impact. You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.

Severity 2

The problem has a *significant* business impact. The program is usable, but it is severely limited.

Severity 3

The problem has *some* business impact. The program is usable, but less significant features (not critical to operations) are unavailable.

Severity 4

The problem has *minimal* business impact. The problem causes little impact on operations, or a reasonable circumvention to the problem was implemented.

Describe problems and gather information

When describing a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Software Support specialists can help you solve the problem efficiently. To save time, know the answers to these questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
- Can you re-create the problem? If so, what steps were performed to re-create the problem?
- Did you make any changes to the system? For example, did you make changes to the hardware, operating system, networking software, and so on.
- Are you currently using a workaround for the problem? If so, be prepared to explain the workaround when you report the problem.

Submit problems

You can submit your problem to IBM Software Support in one of two ways:

Online

Click **Submit and track problems** on the IBM Software Support site at <http://www.ibm.com/software/support/probsub.html>. Type your information into the appropriate problem submission form.

By phone

For the phone number to call in your country, go to the IBM Software Support website support handbook contacts page at <http://www14.software.ibm.com/webapp/set2/sas/f/handbook/contacts.html>,

and click **IBM Directory of worldwide contacts** or select your geographical area for a list of contacts.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Software Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the Software Support website daily, so that other users who experience the same problem can benefit from the same resolution.

Appendix B. Date and time format reference - strftime

Tivoli Workload Scheduler uses the *strftime* standard method for defining the presentation of the date and time in log files generated by CCLog. There is a parameter in the properties file of CCLog, where you define the format (see “Tivoli Workload Scheduler logging and tracing using CCLog” on page 14).

This parameter uses one or more of the following variables, each of which is introduced by a "%" sign, separated, if required, by spaces or other character separators.

For example, to define a date and time stamp that would produce the following (12-hour time, followed by the date) "7:30:49 a.m. - November 7, 2008", you would use the following definition:

```
%l:%M:%S %P - %B %e, %G
```

The full details of the parameters you can use are as follows:

Table 17. strftime date and time format parameters

Parameter	Description	Example
%a	The abbreviated weekday name according to the current locale.	Wed
%A	The full weekday name according to the current locale.	Wednesday
%b	The abbreviated month name according to the current locale.	Jan
%B	The full month name according to the current locale.	January
%c	The preferred date and time representation for the current locale.	
%C	The century number (year/100) as a 2-digit integer.	19
%d	The day of the month as a decimal number (range 01 to 31).	07
%D	Equivalent to %m/%d/%y. (This is the USA date format. In many countries %d/%m/%y is the standard date format. Thus, in an international context, both of these formats are ambiguous and must be avoided.)	12/25/04
%e	Like %d, the day of the month as a decimal number, but a leading zero is replaced by a space.	7
%G	The ISO 8601 year with century as a decimal number. The 4-digit year corresponding to the ISO week number (see %V). This has the same format and value as %y, except that if the ISO week number belongs to the previous or next year, that year is used instead.	2008
%g	Like %G, but without century, i.e., with a 2-digit year (00-99).	04
%h	Equivalent to %b.	Jan
%H	The hour as a decimal number using a 24-hour clock (range 00 to 23).	22
%I	The hour as a decimal number using a 12-hour clock (range 01 to 12).	07
%j	The day of the year as a decimal number (range 001 to 366).	008
%k	The hour (24-hour clock) as a decimal number (range 0 to 23); single digits are preceded by a blank. (See also %H.)	7

Table 17. *strftime date and time format parameters (continued)*

Parameter	Description	Example
%l	The hour (12-hour clock) as a decimal number (range 1 to 12); single digits are preceded by a blank. (See also %I.)	7
%m	The month as a decimal number (range 01 to 12).	04
%M	The minute as a decimal number (range 00 to 59).	58
%n	A newline character.	
%p	Either 'AM' or 'PM' according to the given time value, or the corresponding strings for the current locale. Noon is treated as 'pm' and midnight as 'am'.	AM
%P	Like %p but in lowercase: 'am' or 'pm' or a corresponding string for the current locale.	am
%r	The time in a.m. or p.m. notation. In the POSIX locale this is equivalent to '%I:%M:%S %p'.	07:58:40 am
%R	The time in 24-hour notation (%H:%M). For a version including the seconds, see %T below.	07:58
%s	The number of seconds since the Epoch, i.e., since 1970-01-01 00:00:00 UTC.	1099928130
%S	The second as a decimal number (range 00 to 61). the upper level of the range 61 rather than 59 to allow for the occasional leap second and even more occasional double leap second.	07
%t	A tab character.	
%T	The time in 24-hour notation (%H:%M:%S).	17:58:40
%u	The day of the week as a decimal, range 1 to 7, Monday being 1. See also %w.	3
%U	The week number of the current year as a decimal number, range 00 to 53, starting with the first Sunday as the first day of week 01. See also %V and %W.	26
%V	The ISO 8601:1988 week number of the current year as a decimal number, range 01 to 53, where week 1 is the first week that has at least 4 days in the current year, and with Monday as the first day of the week. See also %U and %W.	26
%w	The day of the week as a decimal, range 0 to 6, Sunday being 0. See also %u.	5
%W	The week number of the current year as a decimal number, range 00 to 53, starting with the first Monday as the first day of week 01.	34
%x	The preferred date representation for the current locale without the time.	
%X	The preferred time representation for the current locale without the date.	
%y	The year as a decimal number without a century (range 00 to 99).	04
%Y	The year as a decimal number including the century.	2008
%z	The time-zone as hour offset from GMT. Required to emit RFC822-conformant dates (using "%a, %d %b %Y %H:%M:%S %Z").	-2
%Z	The time zone or name or abbreviation.	GMT
%%	A literal '%' character.	%

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

Provides information about the trademarks and registered trademarks of IBM and of the companies with which IBM has trademark acknowledgement agreements.

IBM, the IBM logo, and `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>.

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.



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.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation 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

Special characters

@ (atsign) key setup incorrectly on UNIX 98

A

about this guide xiii
access permission problem for Oracle administration user 122
access problems for user on DWC 168
access to Symphony locked by stageman 136
accesses, multiple, from TDWC, wrong user logged in 168
accessibility xiv
action on TDWC, list not updated after running 183
actions return empty tables in TDWC 184
add, command, validating time zone incorrectly 99
administration user, Oracle, access permission problem 122
administrator cannot see navigation tree 169
advanced user rights (incorrect), causing login failure to conman 107
agent
log and trace files 33
agent log and trace files
twstrace syntax 35
agent traces
modifying 34
viewing settings 34
agents
down 95
not linking after first JnextPlan on HP-UX 92
not linking after repeated switchmgr 201
not linking to master domain manager 93
AIX
rmstdlist fails with an exit code of 126 141
An internal error has occurred - AWSJPL006E 137
APARs
IY50132 99
IY50136 14
IY60841 102
application server
creating core dump 51
does not start after keystore password change 122
hanging, creating core dump 51
java process already running 123
log and trace files 37
times out 122
trace settings 37

application server (*continued*)
troubleshooting 122
Applications, Tivoli Workload Scheduler
for, troubleshooting 116
appservman and 8.3 agents 145
at keyword, validating time zone incorrectly 99
audit
database and plan 69
directory for log files 70
dynamic workload scheduling 75
enabling 70
header format 70
initiate 70
log files 70
log format 71
overview 69
restarting to initiate 70
audit trails
maintaining 69
authentication
wrong attempt 165
authentication problem with UpdateStats 117
Autotrace
stopping while running JnextPlan 102
available groups list is empty in enter task information window, using LDAP with TDWC 185
average cpu time, in job statistics view of TDWC, shows -1 171
average duration, in job statistics view of TDWC, shows -1 171
AWKRAA209E 178
AWKRCE012E
connection failure 91
AWSBCV012E received 110
AWSBCW037E received 102
AWSBCW039E received 102
AWSBIA015I received 99
AWSBIA019E received 99
AWSBIA106W received 99
AWSBIA148W received 99
AWSDEC002E received 110
AWSDEQ008E received 115
AWSDEQ024E received 106
AWSECM003E message received 133
AWSEDW001I received 91
AWSEDW020E received 91
AWSITA104E
agents down 95
AWSITA122E 178
AWSITA245E
agents down 95
AWSJCO005E
CORBA NO_PERMISSION 188
AWSJCO084E message issued 117
AWSJCS011E message using planman deploy
not enough space 117

AWSJCS011E message using planman deploy (*continued*)
zip file error 117
AWSJPL017E received 102
AWSMSP104E message, failed mail send 131
AWSUI0924E
problems with import operation 179
AWSUI6171E received 166
AWSUI6182E received 166
AWSWUI0331E error returned from custom SQL query with validate command on TDWC 171

B

background threads continue if browser window closed 175, 185
backup domain manager
agents not linking after repeated switchmgr 201
common problems 200
Symphony file becomes corrupted 201
troubleshooting 195
batchman
fails on a fault-tolerant agent 110
in workload service assurance 192
batchup service fails to start 114
behind firewall, attribute in fault-tolerant agents 91
blank page in DWC
High availability disaster recovery 176, 186
bound z/OS shadow job is carried forward indefinitely 118
browser window closing leaves background threads running 175, 185
built-in troubleshooting features 6

C

can be event processor, used to check workstation event enablement 125
carry forward z/OS bound shadow job never completes 118
ccg_basiclogger, CCLog parameter value 17
ccg_filehandler, CCLog parameter value 17
ccg_multiproc_filehandler, CCLog parameter value 17
ccg_pdlogger, CCLog parameter value 17
CCLog
causing jobs to fail on fault-tolerant agent 109
date and time format 219
description 14
parameters 14, 18

- CCLog (*continued*)
 - performance 18
 - switching 15
- character corruption 147
- CLI
 - for composer
 - gives server access error 98
 - log files 39
 - programs (like composer) do not run 141
- cluster.exe 147
- clusterupg 147
- collected data
 - data capture utility 44
- command
 - startappserver 136
- commands
 - xcli 58
- commands and scripts
 - add, validating time zone
 - incorrectly 99
 - cpuname 91
 - dbexpand, impact on audit log file 73
 - deldep 88
 - evtsize, to enlarge Mailbox.msg file 110
 - makesec, impact on audit log file 73
 - release 88
 - replace, validating time zone
 - incorrectly 99
 - rmstdlist, fails on AIX with an exit code of 126 141
 - rmstdlist, gives different results 141
 - shutdown 201
 - start, not working with firewall 91
 - stop, not working with firewall 91
 - submit job 88
 - submit schedule 88
- completed jobs or job streams not found 143
- composer
 - CLI gives server access error 98
 - display cpu=@ fails on UNIX 98
 - gives a dependency error with interdependent object definitions 97
 - gives the AWSJOM179E error when deleting a workstation 99, 183
 - troubleshooting 97
- Composer deletion of a workstation fails with the AWSJOM179E error 99, 183
- configuration data capture utility 41
- configuration file, event monitoring, empty or missing 133
- conman
 - fails on SLES8 107
 - log type 71
 - login fails on Windows 106
 - troubleshooting 106
- connection from DW
 - error when running historical reports or testing connection from an external instance of WebSphere Application Server 159
 - not working 156

- connection from TDWC
 - fails if Oracle database in use 158
 - fails when performing any operation 159
 - settings not checked 165
 - test, takes several minutes before failing 157
 - troubleshooting 155, 180
- connectivity. troubleshooting 196
- connector
 - refresh after changing auditing parameters 70
- conventions used in publications xiii
- core dump of application server, creating 51
- correlating messages in Log Analyzer 28
- corrupt CSV report generated from TDWC as seen in MS Excel 172
- corrupt Symphony file recovery 205
- corrupt Symphony recovery 207
 - automated procedure 211
 - command-line command 211
 - corrupt Symphony recovery on FTA 211
 - on fault-tolerant agent 211
 - resetFTA command 211
- corrupted
 - Symphony file 208
- cpuname, command 91
- Create Post Reports 139
 - Create Post Reports 139
 - Create Post Reports 139
- credentials error with
 - Tivoli Workload Scheduler for z/OS connector z/OS WebSphere Application Server 170
- critical job
 - high risk, has an empty hot list 194
 - is consistently late 193
- critical network timings changing unexpectedly 193
- critical path and 8.3 agents 145
- critical start times
 - inconsistent 193
 - not aligned 192
- CSV report generated from TDWC is corrupted in MS Excel 172
- custom SQL query returns the error message AWSWUI0331E with validate command on TDWC 171
- customer support 216
- customization
 - CCLog 15

D

- data capture in event of problems 41
- data capture tool
 - used for ffdc 50
- data capture utility 41
 - command syntax 42
 - data collection 44
 - data structure 47
 - parameters 42
 - prerequisites 42
 - syntax 42
 - tasks 43

- data capture utility (*continued*)
 - when to run 41
- database
 - directory for audit files 70
 - log type 71
 - table locked 140
- database and plan audit 69
- database jobs
 - supported JDBC drivers 112
 - troubleshooting 112
- database query returns the error message AWSWUI0331E with validate command on TDWC 171
- database transaction log full on Oracle - JnextPlan fails 121
- date and time format, CCLog
 - parameter 14
 - reference 219
- date errors in jobs
 - time zone incorrect setting 146
- date inconsistency
 - AIX master domain manager 146
- date inconsistency in job streams
 - time zone incorrect setting 146
- daylight saving notation missing in the time zone specification on TDWC (from V8.4 FP1) 186
- db2 179
- DB2
 - deadlock 120
 - error causing JnextPlan to fail 101
 - full transaction log causing JnextPlan to fail 119
 - table locked 140
 - timeout 120
 - times out 119
 - transaction log full 100
 - troubleshooting 119
 - UpdateStats
 - fails after 2 hours 119
- db2 error message 179
- dbexpand, command, impact on audit log file 73
- deadline keyword, validating time zone incorrectly 99
- default tasks not converted into language set in browser, in TDWC 174, 184
- default values in report fields in TDWC after upgrade 173
- deldep, command 88
- delete
 - workstation fails with the AWSJOM179E error 99, 183
- deleted
 - stdlist erroneously 142
- dependencies
 - giving error with interdependent object definitions 97
 - lost when submitting job streams with wildcards 108
 - not processed correctly when enLegacyId is set 135
 - of Job Scheduler instance not updated 143
- deploy (D) flag not set after ResetPlan command used 133

- developer tools
 - freeze panels 177
- diagnostic tools 9
- directories
 - audit 70
 - database 70
 - plan 70
 - pobox, storing messages 88
- disk filling up
 - EDWA 135
- disk usage problems
 - EDWA 135
- display cpu=@ fails, on UNIX 98
- distributed engine
 - responsiveness of TDWC decreasing with 166
 - when running production details reports, might overload TDWC 166
- domain manager
 - agents not linking after repeated switchmgr 201
 - cannot link to fault-tolerant agent 91
 - mailman unlinking from fault-tolerant agents 110
 - not connecting to fault-tolerant agent using SSL 90
 - not shutdown on UNIX after switchmgr 201
 - recovering corrupt Symphony file 205
 - running as standalone 87
 - standalone running 87
 - start and stop, commands not working 91
 - Symphony file on backup becomes corrupted 201
 - UNIX, system processes not killed after switchmgr 201
 - workstation not linking after JnextPlan 103
- domain name
 - not included for mail sender 131
- duplicate user ID invalidating session on TDWC 183
- DWC connection status blank 158
- DWC engine version empty 158
- dynamic agent 152
 - job status continually in running state 94
 - not found from console 94
 - not running submitted job 94
 - server connection 111
 - troubleshooting 111
- dynamic agent (V8.5.1)
 - cannot be registered 111
- dynamic workload broker
 - cached jobs
 - increasing 149
 - concurrent threads on server
 - configuring 149
 - job archiving
 - configuring 149
 - job throughput
 - increasing 149
- Dynamic Workload Console 169, 176, 186

- Dynamic Workload Console (*continued*)
 - access error launching task from bookmark 174, 185
 - accessibility xiv
 - actions return empty tables 184
 - admin user cannot see portfolio 169, 178
 - available groups list is empty in enter task information window, using LDAP 185
 - communication failure with DB2 on RHEL V5.6 179
 - CSV report corrupted in MS Excel 172
 - daylight saving notation missing in the time zone specification 186
 - db2 179
 - default tasks not converted into language set in browser 174, 184
 - engine connection
 - error when running historical reports or testing connection from an external instance of WebSphere Application Server 159
 - fails if Oracle database in use 158
 - fails when performing any operation 159
 - not working 156
 - settings not checked 165
 - test, taking several minutes before failing 157
 - troubleshooting 155, 180
- fields in job statistics view showing -1 171
- insufficient space when running production details reports 172
- internet explorer problem 176, 177
- job stream not displayed 178
- JVM fails on RHEL V5 186
- list not updated after running action 183
- other problems 182
- performance problems 166
- plan view 177
- problems with browsers 174
- problems with reports 170
- processing threads continue in background if browser window closed 175, 185
- production details reports, running, might overload distributed engine 166
- report fields show default values after upgrade 173
- reports not displayed when third party toolbar in use 171
- responsiveness decreasing with distributed engine 166
- run period exceeds the historical data time frame 173
- session has become invalid message received 183
- SQL query returns the error message AWSWUI0331E with validate command 171
- troubleshooting 1, 155

- Dynamic Workload Console (*continued*)
 - unexpected login request when using single sign-on 168
 - unresponsive script warning with Firefox browser when opening Workload Designer 175
 - user access problems 168
 - Workload Designer 178
 - wrong user logged in when making multiple accesses 168
 - WSWUI0331E error when running reports on an Oracle database 172
- Dynamic Workload Console Single Sign-On authentication fails after master upgrade to version 8.6.0 169
- dynamic workload scheduling
 - log files 32
 - trace files 32
- dynamic workload scheduling audit 75

E

- Eclipse
 - installing for Log Analyzer 19
 - prerequisites 19
- education xiv, 213
- edwa and 8.3 agents 145
- EIF event, checking it has been sent 129
- email send action fails
 - for event rule 131
- empty or missing event monitoring configuration file 133
- empty panel in DWC
 - HADR 176, 186
- empty tables returned in TDWC from actions 184
- enDbAudit, global option 70
- enEventDrivenWorkloadAutomation, used to check event management enablement 125
- engine connection from DWC
 - not working 156
- engine connection from TDWC
 - error when running historical reports or testing connection from an external instance of WebSphere Application Server 159
 - fails if Oracle database in use 158
 - fails when performing any operation 159
 - settings not checked 165
 - test, takes several minutes before failing 157
 - troubleshooting 155, 180
- enLegacyId, dependencies not processed correctly 135
- enLegacyStartOfDayEvaluation, time zones not resolving correctly 135
- enPlanAudit, global option 70
- enter task information window, has available groups list empty, using LDAP with TDWC 185
- erroneously deleted stdlist 142
- error AWSJOM179E Composer deletion of a workstation fails 99, 183

- error given with interdependent object definitions 97
- error launching tasks from browser 174, 185
- error messages MakePlan 137, 138
- error messages SwitchPlan 138, 139
- error opening IPC, error message 91
- error opening zip file
 - in planman deploy 117
- error using add task to bookmark, in TDWC 174, 185
- event
 - lost 131
- event counter
 - troubleshooting 195
- event management
 - check if enabled 125
 - checking
 - EIF event has been sent 129
 - FileMonitorPlugIn event has been received 129
 - monconf directory 127
 - that SSM Agent is running 129
 - TWSObjectMonitorPlugIn event has been received 129
 - deploy (D) flag not set after ResetPlan command used 133
 - LogMessageWritten not triggered 132
 - monman deploy messages 128
 - not processed in correct order 134
 - showcpus state values 125
 - troubleshooting 124
 - using getmon 127
- event monitoring configuration file, empty or missing 133
- event processor
 - commands not working 134
 - not deploying rules after switching 132
- event rules
 - do not trigger 124
 - email send action fails 131
 - many, causing planman deploy to fail 134
 - not deployed after switching event processor 132
- evtsize, command to enlarge Mailbox.msg file 110
- Excel showing corrupt CSV report generated from TDWC 172
- exclusive access to Symphony, not possible with stageman 136
- exit code of method substituted for return code (extended agent) 116
- extended agent, troubleshooting 116
- extraneous exception 188

F

- F flag state given for domain manager on UNIX after switchmgr 201
- Failover Cluster Command Interface 147
- fault-tolerant agent
 - cannot link to domain manager 91
 - jobs failing in heavy workload conditions 109

- fault-tolerant agent (*continued*)
 - not connecting to domain manager using SSL 90
 - not linking to master domain manager 93
 - not obeying start and stop commands 91
 - recovering corrupt Symphony file 205
 - running as standalone 87
 - troubleshooting 109
 - unlinking from mailman on domain manager 110
- ffdc
 - See first failure data capture
- FileMonitorPlugIn event, checking it has been received 129
- files
 - localopts, thiscpu option not set correctly 201
 - Mailbox.msg corrupt 110
 - pobox, full 100
 - Sinfonia
 - in recovery of corrupt Symphony file 205
 - to delete after SSL mode change 91
 - Symphony
 - becomes corrupted on backup domain manager 201
 - to delete after SSL mode change 91
 - TWSCCLog.properties 14
- filling percentage of the mailboxes EDWA 135
- final status, jobs or job streams in, not found 143
- Firefox browser giving unresponsive script warning when using the TDWC Workload Designer 175
- firewall, between domain managers 91
- first failure data capture 50
- fix packs
 - keeping up-to-date 7
 - obtaining 215
- fixes 215
- fomatters.basicFmt.dateTimeFormat, CCLog parameter 17
- fomatters.basicFmt.separator, CCLog parameter 17
- forced logout invalidating session on TDWC 183
- format, audit log files 71
- freeze
 - panels 177
- ftbox, troubleshooting 196
- full mailboxes EDWA 135

G

- getmon, used to check workstation monitoring configuration 127
- getting a new socket, error message 91
- glossary xiii
- graphical view problems 177

- groups available list is empty in enter task information window, using LDAP with TDWC 185

H

- hang
 - Windows 105
- hang of application server, creating core dump 51
- header format, audit log records 70
- header, log type 71
- high risk critical job has an empty hot list 194
- highlighting messages in log analyzer 27
- host name
 - not recognized 103
- hot list, empty, for high risk critical job 194
- HP-UX
 - agents not linking after first JnextPlan 92

I

- IBM Redbooks 213
- IBM support assistant 213
- impersonation level errors (Windows) 115
- import settings 179
- In-Flight Trace facility 53
- inconsistent times in planman showinfo 118
- increase job processing 149
- increase processed jobs 149
- information centers
 - at IBM support website, searching for problem resolution 214
 - local, searching for problem resolution 214
- initialization problems 87
- installation
 - Eclipse, for Log Analyzer 19
 - log files 30
- interactive job 152
- interactive job is not visible 152
- interactive jobs not interactive using Terminal Services 113
- internet explorer
 - developer tools 177
 - graphic panels not displayed 176
 - object error 177
- Internet, searching for problem resolution 214
- invalid session message received on TDWC 183
- IP address
 - not recognized 103
- IY50132, APAR 99
- IY50136, APAR 14
- IY60841, APAR 102
- IZ62730 145

J

- J flag state given for domain manager on UNIX after switchmgr 201
- Java compiler error
 - using planman deploy 117
- Java exception
 - not enough space
 - using planman deploy 117
- Java out of memory when running JnextPlan 101
- java process already running restarting Application server 123
- Java Runtime Environment
 - as prerequisite of Eclipse 19
 - fails on TDWC with RHEL V5 186
- java.lang.OutOfMemoryError: PermGen space 178
- java.net.SocketTimeoutException received 166
- JBDC logs
 - activating 32
- JnextPlan
 - CreatePostReports.cmd 105
 - fails
 - AWSJPL017E 102
 - because database log is full 100
 - Java out of memory 101
 - to start 100
 - with DB2 error: nullDSRA0010E: SQL State = 57011, Error Code = -912 101
 - fails because database transaction log full 121
 - fails because DB2 transaction log full 119
 - impact on auditing 70
 - job remains in "exec" status after 103
 - Makeplan.cmd
 - Updatestats.cmd 105
 - not changing available resource quantity in plan 104
 - not initializing remote workstation 102
 - SLES8, after second, agent does not link 105
 - slow 102
 - troubleshooting 99
 - workstation not linking 103
- job
 - bound z/OS shadow, is carried forward indefinitely 118
 - remains in "exec" status 103
- job failure
 - remote command 151
- job number increase 149
- job output character corruption 147
- job rate increase 149
- Job Scheduler instance
 - dependency not updated 143
 - predecessor not updated 143
- Job Scheduler instance mismatch between Symphony and preproduction plan 116
- job shows as not running 94
- job shows as running 94
- job statistics view of TDWC, fields showing -1 171

- job stream view 178
- job streams
 - completed, not found 143
- job types with advanced options
 - database jobs error 112
 - MSSQL jobs error 112
- jobman and JOBMAN
 - fails on a fault-tolerant agent 110
 - in workload service assurance 192
- jobmon and JOBMON
 - fails on a fault-tolerant agent 110
- jobs
 - completed, not found 143
 - failing on fault-tolerant agent in heavy workload conditions 109
 - interactive, not interactive using Terminal Services 113
 - statistics are not updated daily 143
 - with a "rerun" recovery job remains in the "running" state 142

K

- keystore password changed, WebSphere Application Server does not start 122
- knowledge bases, searching for problem resolution 214

L

- L flag state given for domain manager on UNIX after switchmgr 201
- language
 - of log messages 13
- language not being set for default tasks in TDWC 174, 184
- late, consistently, critical job 193
- late, job status, incorrectly reported when time zones not enabled 143
- LDAP
 - account lock 165
- LDAP, using when available groups list is empty in enter task information window (TDWC) 185
- legacy global options, problems using 135
- Limited fault-tolerant agents on IBM i
 - troubleshooting 1
- link problems, troubleshooting 196
- linking
 - agent not found 94
 - no resources available 91, 94
 - problems 88
 - problems with, in dynamic environment 94
 - problems with, on fault-tolerant agent 93
- links
 - cannot be made
 - after SSL mode change 90
 - between fault-tolerant agent and domain manager 91
- links, agents not making after repeated switchmgr 201

- Linux
 - SLES8
 - after second JnextPlan, agent does not link 105
 - conman fails 107
- list not updated after running action on TDWC 183
- local parameters not resolving correctly 144
- localopts
 - merge stdlists 14
 - nm port 91
 - SSL port setting 90
 - thiscpu option not set correctly 201
- locked, database table 140
- locklist problem causing JnextPlan 101
- Log Analyzer
 - adding log file 21
 - analyzing messages with the symptom catalog 29
 - configuring memory usage 19
 - description 18
 - Eclipse 19
 - installing plug-in 20
 - installing symptom catalog 29
 - installing TPTP 19
 - messages
 - correlating 28
 - filtering 24
 - following the flow 24
 - highlighting 27
 - locating 24
 - properties, managing 26
 - sorting 24
 - reports, creating 26
 - symptom catalog 28
 - understanding main window 23
 - upgrading 20
 - using 22
- log and trace files
 - agent 33
- log file
 - content 30
 - location 30
- log files
 - adding to Log Analyzer 21
 - audit
 - format 71
 - location 70
 - command-line client 39
 - database, full 100
 - for application server 37
 - location 14
 - question marks found in 141
 - separate from trace files 13
- logging
 - dynamic workload scheduling 32
 - engine log file switching 15
 - file locations 14
 - modify logging level (quick reference) 9
 - overview 9
- login request, unexpected, when using single sign-on 168
- login to conman fails on Windows 106
- LogMessageWritten event not triggered 132

logout (forced) invalidating session on
TDWC 183
low disk space
EDWA 135

M

Mailbox.msg
file, corrupt 110
mailman
fails on a fault-tolerant agent 110
message from, stops event
counter 196
messages
when SSL connection not
made 90
no incoming message from 110
mailSenderName option
not defined 131
makeplan 136
MakePlan problems 137, 138
makesec
impact on audit log file 73
log type 71
manage roles, administrative user not
authorized 169
master domain manager
recovering corrupt Symphony
file 205
memory
problem, Java, when running
JnextPlan 101
messages
analyzing in Log Analyzer 29
concerning ftbox on full-status
agent 196
from mailman, stopping event
counter 196
from writer, stopping event
counter 195
log, described 13
not being tracked 195
trace, described 13
xcli 65
method exit code substituted for return
code (extended agent) 116
mismatch of Job Scheduler instances
between Symphony and preproduction
plan 116
missing or empty event monitoring
configuration file 133
mixed version environments
workaround 145
modifying
agent traces 34
monconf directory, checking for
monitoring configuration
availability 127
monman and 8.3 agents 145
monman deploy messages 128
MS Excel showing corrupt CSV report
generated from TDWC 172
MSSQL jobs
supported JDBC drivers 112
troubleshooting 112
multiple accesses from TDWC, wrong
user logged in 168

N

netman
two instances listening on the same
port 91
network
common problems 89
link problems 88
problems, common 89
recovery 87
troubleshooting 87
network timings, critical, changing
unexpectedly 193
nm port, localopts parameter 91
nullDSRA0010E error causing JnextPlan
to fail 101

O

Onnnn.hhmm files
deleting 146
opening Workload Designer from
graphical view with Firefox 176
optman, enabling audit 70
Oracle
transaction log full 100
troubleshooting 121
Oracle database giving WSWUI0331E
error when running reports in
TDWC 172
order of events not respected 134
organization parameter, in CCLog 14

P

panel hangs on shared was 178
panels not displayed with internet
explorer 176
parameters, local, not resolving
correctly 144
parms, log type 71
parms, not resolving local parameters
correctly 144
performance
CCLog 18
logging 18
troubleshooting for TDWC 166
performance - troubleshooting 85
permissions problem for Oracle
administration user 122
plan directory for audit files 70
plan monitor
in workload service assurance 191
plan, log type 71
planman
deploy, failing with many rules 134
planman deploy
fails with Java compiler error 117
insufficient space error 117
planman showinfo displays inconsistent
times 118
planner
in workload service assurance 191
troubleshooting 116
plug-in deploy
fails with Java compiler error 117
plug-in, Eclipse, for Log Analyzer 20

plug-in, Eclipse, for Log Analyzer
message help 29
pobox
directory, storing messages 88
file, full 100
post-uninstallation clean up 146
predecessor to Job Scheduler instance
not updated 143
preproduction plan has different Job
Scheduler instances than Symphony
file 116
problem determination
describing problems 217
determining business impact 217
submitting problems 217
problem resolution 213
problems MakePlan 137, 138
problems SwitchPlan 138, 139
problems, other, on TDWC 182
processing threads continue in
background if browser window
closed 175, 185
product
parameter, in CCLog 14
production details reports run in TDWC,
insufficient space to complete 172
production details reports, running from
TDWC, might overload distributed
engine 166
prompts, duplicate numbers 108
publications xiii

Q

question marks found in the stdlist 141

R

recover
corrupt Symphony 208
recovering
corrupt Symphony file 205
network failures 87
recovering a corrupt Symphony file 207
Red Hat Enterprise Linux
V5, JVM failing when using
TDWC 186
V5.6, access to preferences repository
failing 179
Redbooks, IBM 213
release, command 88
release, log type 71
remote command 151
remote command job
connection failure 91
remote workstation not initializing after
JnextPlan 102
replace, command, validating time zone
incorrectly 99
replay protocol, after switchmgr 195
report fields show default values in
TDWC after upgrade 173
report problems, on DWC 174
report problems, on TDWC 170

- reports getting WSWUI0331E error when running on an Oracle database in TDWC 172
- reports not displayed in TDWC when third party toolbar in use 171
- reports, not including completed jobs or job streams 143
- rerun recovery job, original job remains in the "running" state 142
- ResetPlan command
 - not setting deploy (D) flag 133
- resource quantity changes in database not also implemented in plan after JnextPlan 104
- responsiveness of TDWC decreasing with distributed engine 166
- return codes, unrecognized (extended agent) 116
- rights problem for Oracle administration user 122
- rmstdlist command
 - fails on AIX with an exit code of 126 141
 - gives different results 141
- rules (event)
 - do not trigger 124
- rules deploy
 - insufficient space error 117
- run time
 - log files 30
- runmsgno, reset 108
- running but not visible on dynamic agent 152
- running state, original job remains in, with a "rerun" recovery job 142

S

- sample audit log entries 74
- scratch option
 - in planman deploy
 - insufficient space 117
- Security Enhanced Linux, making access to preferences repository fail on TDWC with RHEL V5.6 179
- Security Enhanced Linux, making JVM fail on TDWC with RHEL V5 186
- SELinux, making access to preferences repository fail on TDWC with RHEL V5.6 179
- SELinux, making JVM fail on TDWC with RHEL V5 186
- separator parameter, in CCLog 14
- service pack (Windows), problems after upgrading with 115
- services (Windows)
 - fail to start 113, 114
 - Tivoli Token Service, causing login failure to conman 106
- session has become invalid message received on TDWC 183
- setting trace levels for application server 37
- settings
 - agent traces 34
- shadow bound z/OS job is carried forward indefinitely 118

- showinfo (planman) displays 118
- ShutDown_clu.cmd 147
- shutdown, command 201
- Sinfonia, file
 - in recovery of corrupt Symphony file 205
 - to delete after SSL mode change 91
- Single Sign-On authentication fails after master upgrade 169
- single sign-on, unexpected login request received 168
- SLES8
 - after second JnextPlan, agent does not link 105
 - agent, conman fails 107
- SocketTimeoutException received 166
- software support 213
- Software Support
 - contacting 216
 - describing problems 217
 - determining business impact 217
 - receiving weekly updates 215
 - submitting problems 217
- solaris, shared was cause view 178
- space insufficient when running
 - production details reports in TDWC 172
- Special characters
 - corruption 147
- SQL query returns the error message AWSWUI0331E with validate command on TDWC 171
- SSL
 - no connection between fault-tolerant agent and its domain manager 90
 - port setting in localopts 90
 - workstation cannot link after changing mode 90
- SSM Agent, checking for event processing 129
- stageman, log type 72
- stageman, unable to get exclusive access to Symphony 136
- standalone mode for fault-tolerant agents and domain managers 87
- standard list file 14
- start times, critical
 - inconsistent 193
 - not aligned 192
- start-of-plan-period problems 87
- start, command not working with
 - firewall 91
- startappserver
 - command 136
- Startup_clu.cmd 147
- statistics are not updated daily 143
- status of TWS processes
 - EDWA 135
- stdlist
 - erroneously deleted 142
 - restricting access to 13
- stdlist, question marks found in 141
- stop, command
 - not working with firewall 91
- stopeventprocessor, not working 134
- strftime (date and time format) 219

- structure of extracted data
 - data capture utility 47
- submit job streams with wildcards loses dependencies 108
- submit job, command 88
- submit schedule, command 88
- substitution of return code for method exit code (extended agent) 116
- support 213
- support assistant 213
- support website, searching to find software problem resolution 214
- swap space problem
 - using planman deploy 117
- switcheventprocessor, not working 134
- switching logs in CCLog 15
- switchmgr
 - used repeatedly 201
- switchmgr command, UNIX system processes not being killed after 201
- SwitchPlan problems 138, 139
- Symphony corruption 207
- Symphony file
 - becomes corrupted on backup domain manager 201
 - corrupt 205
 - corrupted 208
 - different Job Scheduler instances than preproduction plan 116
 - managing concurrent access to 136
 - recovery 205
 - to delete after SSL mode change 91
 - troubleshooting 205
- Symphony recovery 207
- symptom catalog, used in Log Analyzer 28
- system resources scan
 - notify scan 112
 - troubleshooting 112
- systemout exeption 188

T

- table, database, locked 140
- task information entry window, has available groups list empty, using LDAP with TDWC 185
- TDWC test connection failure 160, 161, 162
- TDWC Workload Designer does not show on foreground with Firefox browser 176
- technical training xiv
- Terminal Services, interactive jobs not interactive when using 113
- Test and Performance Tools Platform, installation 19
- test connection to engine from TDWC takes several minutes before failing 157
- The database is already locked - AWSJPL018E 137
- the specified run period exceeds the historical data time frame 173
- thiscpu option not set correctly in localopts file 201

- threads continue in background if
 - browser window closed 175, 185
 - time and date format, CCLog
 - parameter 14
 - reference 219
 - time errors in jobs
 - time zone incorrect setting 146
 - time inconsistency
 - AIX master domain manager 146
 - time inconsistency in job streams
 - time zone incorrect setting 146
 - time zone
 - not enabled, causing time-related status problems 143
 - not recognized by WebSphere Application Server 118
 - not validated correctly by composer 99
 - summer notation missing on TDWC (from V8.4 FP1) 186
 - time zones, not resolving when
 - enLegacyStartOfDayEvaluation is set 135
 - time-related status
 - incorrect when time zone not enabled 143
 - timeout
 - on WebSphere Application Server 119
 - while running DB2 UpdateStats job 119
 - timeout of session on TDWC 183
 - timeout on application server 122
 - timeout on DB2 119
 - times inconsistent in planman
 - showinfo 118
 - timings, network, critical, changing unexpectedly 193
 - Tivoli technical training xiv
 - Tivoli Token Service
 - causing login failure to conman 106
 - fails to start 113
 - Tivoli Workload Automation
 - overview 1
 - Tivoli Workload Dynamic Broker
 - troubleshooting 1
 - Tivoli Workload Scheduler for Applications
 - troubleshooting 1
 - Tivoli Workload Scheduler for Applications, troubleshooting 116
 - Tivoli Workload Scheduler for Virtualized Data Centers
 - troubleshooting 1
 - Tivoli Workload Scheduler for z/OS
 - troubleshooting 1
 - Tivoli Workload Scheduler service for TWS_user
 - fails to start 113, 114
 - toolbar, third party, stopping display of reports in TDWC 171
 - tools
 - CCLog 14
 - database and plan 69
 - dynamic workload scheduling 75
 - Log Analyzer 18
 - tools, for troubleshooting 9
 - TOS errors, on fault-tolerant agent 109
 - TPTP 19
 - trace and log files
 - agent 33
 - trace and log files agent
 - agent twstrace syntax 35
 - trace file
 - activation 31
 - trace files
 - for application server 37
 - question marks found in 141
 - separate from log files 13
 - trace information
 - gathering 41
 - trace levels
 - application server setting 37
 - tracing
 - dynamic workload scheduling 32
 - modify logging level (quick reference) 9
 - overview 9
 - tracing facility 53
 - trademarks 222
 - training
 - technical xiv
 - transaction log for database is full 100
 - transaction log for the database is full
 - message received from DB2, causing JnextPlan to fail 119
 - troubleshooting 136
 - application server 122
 - built-in features 6
 - common problems 97
 - composer 97
 - concurrent accesses to the Symphony file 136
 - conman 106
 - database jobs 112
 - DB2 119
 - DWC
 - database 179
 - graphical view 177
 - problems with browsers 174
 - user access problems 168
 - dynamic agent 111
 - dynamic agent (V8.5.1) 111
 - event management 124
 - extended agents 116
 - fault-tolerant agents 109
 - fault-tolerant switch manager 195
 - finding information in other manuals 1
 - JnextPlan 99
 - legacy global options 135
 - Limited fault-tolerant agents on IBM i 1
 - miscellaneous problems 139
 - MSSQL jobs 112
 - networks 87
 - Oracle 121
 - performance 85
 - planner 116
 - Symphony file corruptions 205
 - system resources scan 112
 - TDWC 155
 - engine connections 155, 180
 - troubleshooting (*continued*)
 - TDWC (*continued*)
 - other problems 182
 - performance problems 166
 - problems with reports 170
 - Tivoli Workload Dynamic Broker 1
 - tools 9
 - TWS for Applications 1
 - TWS for Virtualized Data Centers 1
 - TWS for z/OS 1
 - Windows 113
 - workload service assurance 191
 - TWS_user
 - unable to login to conman 106
 - tw.loggers.className, CCLog
 - parameter 17
 - tw.loggers.msgLogger.level, CCLog
 - parameter 16
 - tw.loggers.organization, CCLog
 - parameter 17
 - tw.loggers.product, CCLog
 - parameter 18
 - tw.loggers.trc<component>.level, CCLog
 - parameter 16
 - TWSCCLog.properties
 - customization 15
 - TWSCCLog.properties, file 14
 - twshnd.logFile.className, CCLog
 - parameter 17
 - TWSObjectMonitorPlugIn event, checking it has been received 129
 - twstrace syntax
 - agent log and trace files 35
- ## U
- Unable to access to preferences repository on TDWC with RHEL V5.6 179
 - UNIX
 - display cpu=@ fails 98
 - rmstdlist, fails on AIX with an exit code of 126 141
 - rmstdlist, gives different results 141
 - system processes not killed on ex domain manager after switchmgr 201
 - unlinking
 - fault-tolerant agents from mailman on domain manager 110
 - unrecognized return code (extended agent) 116
 - unresponsive script warning with Firefox browser when using the TDWC Workload Designer 175
 - until keyword, validating time zone incorrectly 99
 - Update Stats 139
 - Update Stats 139
 - Update Stats 139
 - UpdateStats fails after 2 hours 119
 - UpdateStats, fails if longer than two hours 117
 - upgrade
 - Windows, problems after 115
 - your whole environment 7
 - upgrade, making report fields show default values in TDWC 173

- user access problems, on DWC 168
- user, wrong, logged in when making multiple accesses from TDWC 168
- users
 - not authorized to access server, error given by CLI programs 141
 - rights
 - causing login failure to conman 107
 - Windows, problems with 115
 - TWS_user
 - unable to login to conman 106

V

- validate command returns the error message AWSWUI0331E from TDWC database query 171
- validation error given with interdependent object definitions 97
- variable tables
 - default not accessible 144
- variables
 - not resolved after upgrade 144, 147
- viewing
 - agent traces 34
- virtual memory problem
 - using planman deploy 117

W

- web page error with internet explorer 177
- WebSphere startup failure with LDAP 163, 164
- Windows
 - conman login fails 106
 - Terminal Services, interactive jobs not interactive when using 113
 - troubleshooting 113
 - upgrading, problems after 115
 - user rights, problems with 115
- workload
 - fault-tolerant agent causing jobs to fail 109
- workload designer does not open 170
- workload service assurance
 - critical job
 - is consistently late 193
 - critical network timings changing unexpectedly 193
 - critical start times
 - inconsistent 193
 - not aligned 192
 - high risk critical job has an empty hot list 194
 - troubleshooting 191
 - use of batchman 192
 - use of jobman 192
 - use of plan monitor 191
 - use of planner 191
- workstation not linking after JnextPlan 103
- workstations
 - not linking after JnextPlan 103

- workstations (*continued*)
 - not shutdown on UNIX after switchmgr 201
 - remote, not initializing after JnextPlan 102
- writer
 - message from, stops event counter 195
 - messages
 - when SSL connection not made 90
- wrong schedtime in jobs
 - time zone incorrect setting 146
- wrong start time in jobs
 - time zone incorrect setting 146
- WSWUI0331E error when running reports on an Oracle database in TDWC 172

X

- xcli 53
 - messages 65
- xcli command 58
- xtrace.ini
 - description 55
 - modify 56
 - syntax 56

Z

- z/OS bound shadow job is carried forward indefinitely 118



Product Number: 5698-WSH

Printed in USA

SC32-1275-14

