

Tivoli Workload Scheduler
Version 8.6

Integrating with Other Products



Tivoli Workload Scheduler
Version 8.6

Integrating with Other Products



Note

Before using this information and the product it supports, read the information in Notices.

This edition applies to version 8, release 6, modification level 0 of IBM Tivoli Workload Scheduler (program number 5698-WSH) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC23-8904-03.

© **Copyright IBM Corporation 2007, 2011.**

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

Contents

Figures v

About this publication vii

What is new in this release	vii
What is new in this publication	vii
What this guide contains	vii
Publications	viii
Accessibility	viii
Tivoli technical training	viii
Support information	viii

Chapter 1. Integrating with IBM Tivoli

Monitoring 1

Integrating with IBM Tivoli Monitoring using a script data provider	1
Resources monitored in Tivoli Workload Scheduler	4
Integrating with Tivoli Enterprise Portal using a file data provider	9
Integrating Tivoli Workload Scheduler with Tivoli Enterprise Portal	9
Integrating dynamic workload broker with Tivoli Enterprise Portal	19
Scenario: integrating with Tivoli Enterprise Portal	27
Running the scenario	34

Chapter 2. Integrating with Tivoli

NetView 37

Tivoli Workload Scheduler/NetView overview	37
Installing the integration software	37
Installing on managed nodes and NetView clients	38
Installing on the management node or NetView server	38
Setting up Tivoli Workload Scheduler/NetView	39
Menu actions	40
Events	41
Polling and SNMP traps	43
Tivoli Workload Scheduler/NetView configuration files	44
The BmEvents configuration file	44
The MAgent configuration file	45
Monitoring writers and servers	47
Tivoli Workload Scheduler/NetView configuration options	47
Agent scan rate	47
Manager polling rate	47

Configuring agents in NetView	48
Configuring workstation status in NetView	48
Unison software MIB	48
Re-configuring enterprise-specific traps	48
Tivoli Workload Scheduler/NetView program reference	50
mdemon synopsis	50
magent synopsis	51

Chapter 3. Integrating with Tivoli

Enterprise Console 53

Configuring the Tivoli Enterprise Console adapter	53
Configuring the Tivoli Enterprise Console server	54
Event formats	55
Positional event variables	56
Re-loading monitoring data	59
Job scheduling events	61
Job scheduling events format	69

Chapter 4. Integrating with Tivoli

License Compliance Manager 73

Chapter 5. Integrating with Tivoli

Storage Manager 75

Chapter 6. Integrating with the Change and Configuration Management

Database 83

Managing RFCs using the integration between Tivoli Workload Scheduler and Change and Configuration Management Database	87
Integrating dynamic workload broker with Configuration Management Database	88

Chapter 7. Integrating dynamic workload broker with Tivoli

Provisioning Manager 93

Notices 95

Trademarks	96
----------------------	----

Index 99

Figures

1.	Tivoli Workload Scheduler situations	3	5.	IBM Tivoli Monitoring architecture.	10
2.	Situation content	3	6.	Information flow.	29
1 3.	Enterprise status view	4	7.	Event generation flow	62
1 4.	Tivoli Workload Scheduler monitored		8.	Integration environment	87
1	applications	6			

About this publication

This publication documents integration scenarios to show how you can integrate Tivoli Workload Scheduler with other Tivoli® products.

What is new in this release

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

For information about the APARs that this release addresses, see the Tivoli Workload Scheduler Download Document at <http://www.ibm.com/support/docview.wss?rs=672&tuid=swg24027501>.

What is new in this publication

Apars and defects have been fixed.

All changes are marked with revision bars.

What this guide contains

This manual contains the following chapters and appendixes:

- Chapter 1, “Integrating with IBM Tivoli Monitoring,” on page 1
Describes how Tivoli Workload Scheduler integrates with Tivoli Monitoring version 6.1.
- Chapter 2, “Integrating with Tivoli NetView,” on page 37
Describes how Tivoli Workload Scheduler on UNIX integrates with Tivoli NetView® for AIX® version 7.1x.
- Chapter 3, “Integrating with Tivoli Enterprise Console,” on page 53
Describes how Tivoli Workload Scheduler integrates with Tivoli Enterprise Console versions 3.8 and 3.9.
- Chapter 4, “Integrating with Tivoli License Compliance Manager,” on page 73
Describes how Tivoli Workload Scheduler integrates with Tivoli License Compliance Manager version 2.3.
- Chapter 5, “Integrating with Tivoli Storage Manager,” on page 75
Describes how Tivoli Workload Scheduler integrates with Tivoli Storage Manager version 5.4 and later.
- Chapter 6, “Integrating with the Change and Configuration Management Database,” on page 83
Describes what you can achieve by implementing the integration between Tivoli Workload Scheduler and Change and Configuration Management Database (CCMDB).
- Chapter 7, “Integrating dynamic workload broker with Tivoli Provisioning Manager,” on page 93

Describes how to use dynamic workload broker and Tivoli Provisioning Manager together to dynamically create the resources necessary to run workload in your IT environment.

Publications

Full details of Tivoli Workload Automation 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 *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. Integrating with IBM Tivoli Monitoring

How Tivoli Workload Scheduler integrates with Tivoli Monitoring.

This chapter describes how Tivoli Workload Scheduler integrates with IBM Tivoli Monitoring 6.1. You can integrate Tivoli Workload Scheduler with Tivoli Monitoring 6.1 Fix Pack 5 using data providers that are the interfaces of the IBM Tivoli Universal Agent. They enable the collection of data from data sources, client programs, URLs, scripts, relation tables, or SNMP agents and the transfer of the collected data to the IBM Tivoli Universal Agent, an agent that monitors any collected data.

Depending on the type and location of the data you want to monitor, you can use a different data provider and have the following integrations:

- If you use the script data provider, you can enable an integration with Tivoli Monitoring that monitors Tivoli Workload Scheduler data by running a script at regular intervals and parsing the script output to look for errors, anomalies, and threshold conditions. For details about this integration, see “Integrating with IBM Tivoli Monitoring using a script data provider.”
- If you use the file data provider you can enable an integration with Tivoli Enterprise Portal, a Tivoli Monitoring component. This data provider monitors data in a sequential text file. For details about this integration, see “Integrating with Tivoli Enterprise Portal using a file data provider” on page 9.

Note: All .TAR and BAROC files for the Tivoli Monitoring integration are located on the installation DVD in the **integrations\ITM** directory.

Integrating with IBM Tivoli Monitoring using a script data provider

The main prerequisite for this integration is the installation of the Universal Agent on the same workstation where the Tivoli Workload Scheduler instance to be monitored is installed.

Tivoli Workload Scheduler installation provides you with two scripts used as data providers to configure the Universal Agent that monitors Tivoli Workload Scheduler. For information about the operating systems supported by this integration, refer to the IBM Tivoli Monitoring 6.1 documentation.

During the installation of Tivoli Workload Scheduler, the following configuration scripts are copied into the ITM directory of the Tivoli Workload Scheduler workstation to be monitored:

- ITMconfig.sh for UNIX and ITMconfig.cmd for Windows

This script configures the Universal Agent to monitor Tivoli Workload Scheduler. The syntax is the following:

```
ITMconfig -ua_home_dir ua_home_dir
```

where:

-ua_home_dir

Specifies the home directory of the Universal Agent. This script runs on the workstation where the Universal Agent is installed.

Note: Make sure that the user running the Tivoli Universal Agent has read access to the pobox directory.

- `ITMCreateSituations.sh` for UNIX and `ITMCreateSituations.cmd` for Windows
This script configures the Tivoli Enterprise Portal Console to create and display the Tivoli Workload Scheduler situations. Copy this script, the `baseNameSituation.xml` file, and `ITM_TWS_84metafile_win.mdl` for Windows or `ITM_TWS_84metafile_ux.mdl` for UNIX, onto the Tivoli Enterprise Monitoring Server and run it using the following syntax:

```
ITMCreateSituations -user username -password passwd -host host -ITMHome ITMHome [-TWShome TWShome]
```

where:

username

Specifies the user ID of the IBM Tivoli Monitoring user.

passwd Specifies the password of the IBM Tivoli Monitoring user.

host Specifies the host name of the Tivoli Enterprise Monitoring Server.

ITMHome

Specifies the home directory of the Tivoli Enterprise Monitoring Server.

TWShome

Specifies the home directory of the Tivoli Workload Scheduler. Use this option only if, on the workstation where the Tivoli Enterprise Portal is installed, Tivoli Workload Scheduler is also installed.

Note: If Tivoli Workload Scheduler version 8.5 is installed as an upgrade to a Tivoli Workload Scheduler version 8.4.x instance that is already configured for integration with IBM Tivoli Monitoring, perform the following steps before running the `ITMCreateSituation` script. This ensures that IBM Tivoli Monitoring situations are created correctly.

1. Delete the situations created with the Tivoli Workload Scheduler version 8.4 fix pack 1 integration with IBM Tivoli Monitoring. See the documentation for IBM Tivoli Monitoring 6.1.
2. Edit the `ITMCreateSituations.sh` or `ITMCreateSituations.bat` scripts installed with the Tivoli Workload Scheduler version 8.5 upgrade. Change any entry with the name `CUSTOM_ITM_TWS_APPL00` to `CUSTOM_ITM_TWS_APPL01`. Save the files to the Tivoli Enterprise Monitoring Server.

These steps ensure that the situations are created for the correct Tivoli Workload Scheduler application name. In this case, it is `ITM_TWS_APPL01`.

When you log on to Tivoli Enterprise Portal for the first time, you find the following situations:

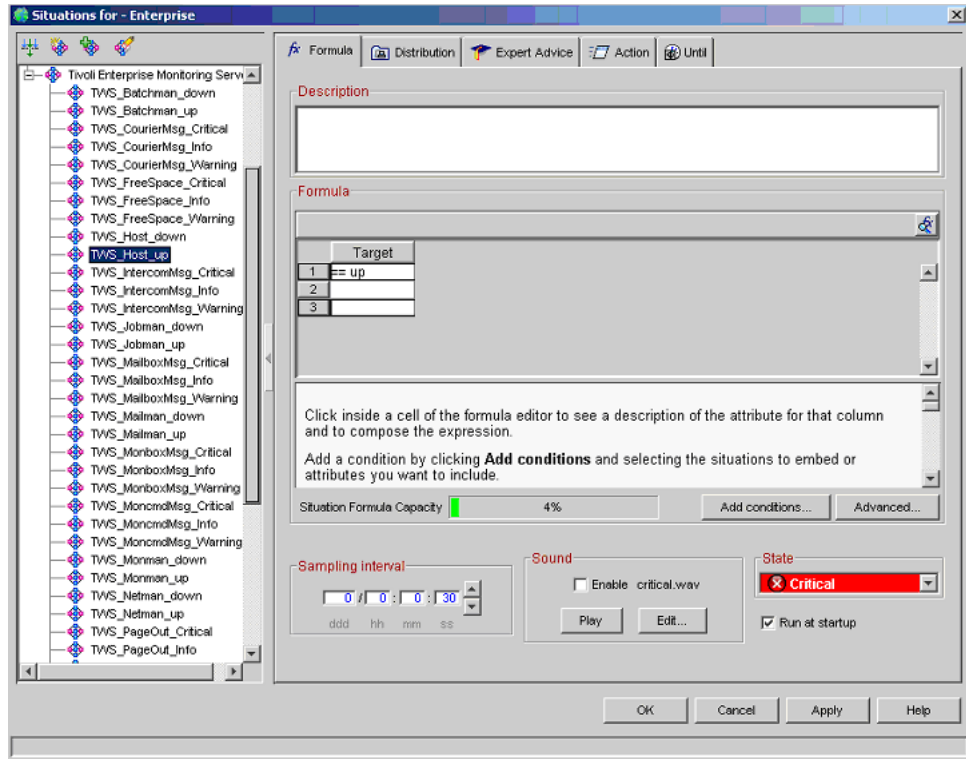


Figure 1. Tivoli Workload Scheduler situations

After the creation of the Tivoli Workload Scheduler situations, in the Tivoli Enterprise Portal Desktop you can see the events associated to each situation:

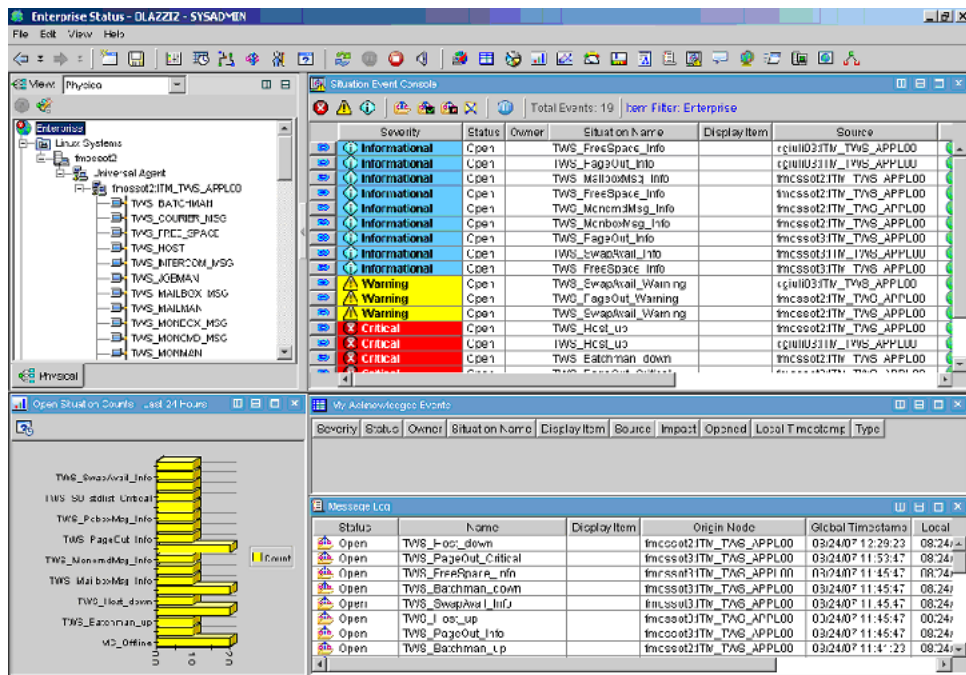


Figure 2. Situation content

Note: The time-to-live <TTL> value, indicating the time that the monitored data will be kept by the Tivoli Enterprise Portal, is set to 86400 seconds (default

value of 1 day). To change this value, you must manually edit the universal agent metafile: ITM_TWS_84metafile_win.mdl (Windows environments) or ITM_TWS_84metafile_ux.mdl (UNIX environments). For further information refer to the Tivoli Universal Agent User's Guide.

As shown by the following figure, the status window for every process you select, reports:

- The selected process name
- The status
- The local time stamp

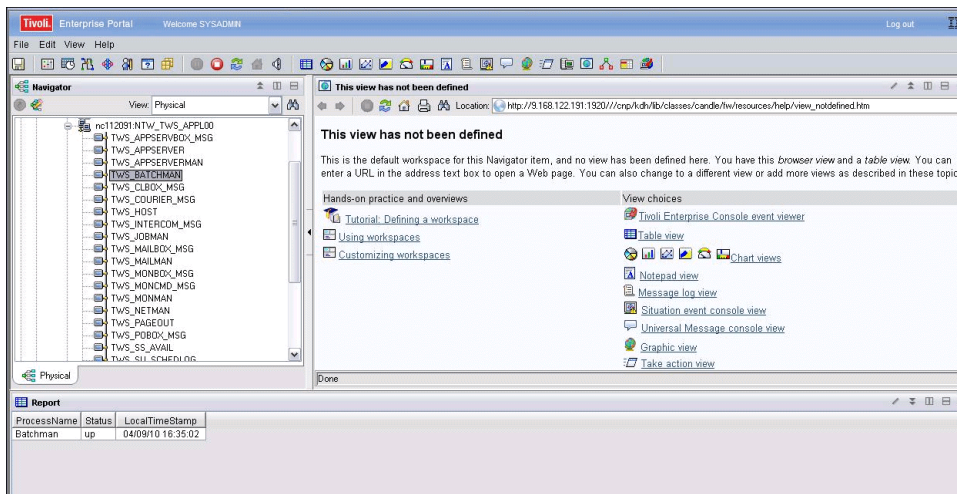


Figure 3. Enterprise status view

Resources monitored in Tivoli Workload Scheduler

The following resources are monitored by Tivoli Universal Agent:

- “Host availability”
- “Application status” on page 5
- “Tivoli Workload Scheduler used space” on page 6
- “Tivoli Workload Scheduler available disk space” on page 6
- “Page-outs” on page 7
- “Swap space available” on page 7
- “Free space” on page 7
- “Tivoli Workload Scheduler Message file size” on page 8

Host availability

The Universal Agent monitors if the parent workstation that can respond to a ping request is available on the network. By default this check is made every hour.

The following table lists the pre-configured actions for this monitoring resource:

Table 1. Events formats table

Severity	Trigger When	Default Actions
Critical	The parent workstation becomes unavailable	Send event

Table 1. Events formats table (continued)

Severity	Trigger When	Default Actions
Informational	The parent workstation becomes available	Send event

Application status

Every minute, the Universal Agent monitors if a process is up and running.

By default the following scheduling processes are monitored:

1

- appserverman
- batchman
- jobman
- mailman
- monman
- netman
- WebSphere Application Server

1

These are the versions of the application status monitor:

- JOBMAN for Windows
- jobman for UNIX

The following table lists the pre-configured actions for this monitoring resource:

1

Table 2. Events formats table

1

Severity	Trigger When	Default Actions
Critical	If one of the monitored processes becomes unavailable (labelled as DOWN on ITM)	Send event
Informational	If one of the monitored processes becomes available (labelled as UP on ITM)	Send event
Warning	If one of the monitored processes is stopped by a user (labelled as STOP on ITM)	Send event

1

1

1

1

1

1

1

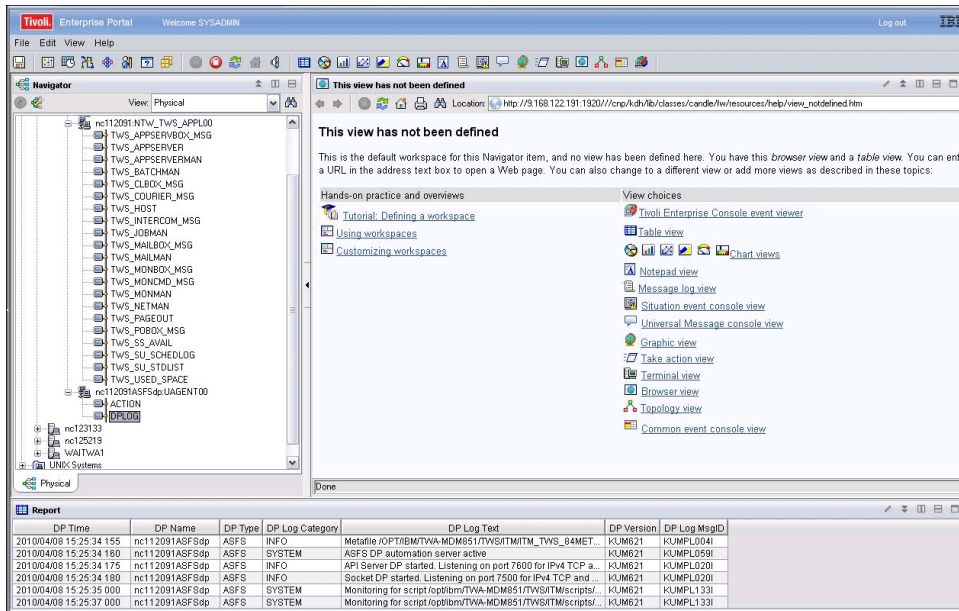
1

1

1

The next figure shows the list of Tivoli Workload Scheduler monitored applications:

1



1

1 Figure 4. Tivoli Workload Scheduler monitored applications

Tivoli Workload Scheduler used space

Every 15 minutes, the Universal Agent monitors the number of megabytes used by a specific directory. The default Tivoli Workload Scheduler directories are:

- stdlist
- schedlog

The following table lists the pre-configured actions for this monitoring source:

Table 3. Events formats table

Severity	Trigger When	Default Actions
Critical	The used space is greater than 307200 Kilobytes.	Send event
Warning	The used space value is between 102401 and 307200 Kilobytes.	Send event
Informational	The used space is less than or equal to 102400 Kilobytes.	Send event

Tivoli Workload Scheduler available disk space

Every 15 minutes, the Universal Agent monitors the amount of free disk space in the Tivoli Workload Scheduler directory.

The following table lists the pre-configured actions for this monitoring resource:

Table 4. Events formats table

Severity	Trigger When	Default Actions
Critical	The free disk space is less than 2 MB.	Send event
Warning	The free disk space is less than 6 MB.	Send event

Table 4. Events formats table (continued)

Severity	Trigger When	Default Actions
Informational	The free disk space is greater than 6 MB.	Send event

Page-outs

Every five minutes, the Universal Agent monitors the number of kilobytes per second paged out on a workstation.

The following table lists the pre-configured actions for this monitoring resource:

Table 5. Events formats table

Severity	Trigger When	Default Actions
Critical	Kilobytes per second paged out are greater than 100 KB.	Send event
Warning	Kilobytes per second paged out are greater than 50 KB.	Send event
Informational	Kilobytes per second paged out are less than 50 KB.	Send event

Swap space available

Every 15 minutes, the Universal Agent monitors the amount of available swap space.

The following table lists the pre-configured actions for this monitoring source:

Table 6. Events formats table

Severity	Trigger When	Default Actions
Critical	The swap space is less than 10 MB.	Send event
Warning	The swap space is less than 25 MB.	Send event
Informational	The swap space is greater than 25 MB.	Send event

Free space

Every 15 minutes, the Universal Agent monitors the amount of free space of the file system that mounts the Tivoli Workload Scheduler directory.

The following table lists the pre-configured actions for this monitoring source:

Table 7. Events formats table

Severity	Trigger When	Default Actions
Critical	The file system free space is less than 5%.	Send event
Warning	The file system free space is less than 50%.	Send event
Informational	The file system free space is greater than 90%.	Send event

Tivoli Workload Scheduler Message file size

Every 15 minutes, the Universal Agent monitors the number of megabytes used by a specific message file and can send an event if the percentage written of the size allocated exceeds a predefined limit. By default, the following Tivoli Workload Scheduler files are monitored:

- Appserverbox.msg
- Clbox.msg
- Courier.msg
- Intercom.msg
- Mailbox.msg
- Monbox.msg
- Moncmd.msg
- Pobox files
- All files in the Pobox directory

Note: Because the Pobox can contain message files for different Tivoli Workload Scheduler instances, and the limit is set for the whole directory, the event will show the name of the largest file in the directory.

To be able to monitor the Pobox files with accuracy, run the following steps:

1. Edit the TWSPoBoxSize.cmd (.sh) script and set the TWS_HOME keyword to the name of the Tivoli Workload Scheduler installation directory.
2. Open the ITM_TWS_84metafile_win.mdl (ITM_TWS_84metafile_ux.mdl) metafile and set the SOURCE SCRIPT keyword to the full path of the TWSPoBoxSize.cmd (.sh) script.
3. Run the ITMConfig script to refresh the metafile.

Note that the metafile includes a new attribute named Filename. For example:

```
//NAME TWS_Pobox_msg K 86400 AddTimeStamp
//SOURCE SCRIPT 'C:\IBM\TWA\TWS\ITM\scripts\TWSPoboxSize.cmd' Interval=30
//Attributes
  Filename      D 2048 KEY ATOMIC
  Percentage    C 100
```

4. Check if another application is created.

If this is the case, you need to delete the old situations and rerun the ITMCreateSituation.cmd (.sh) script after you have replaced the new application.

Note: The Filename attribute uses the key atomic options. This means that you can edit a Pobox-related situation (for example TWS_PoboxMsg_Critical). To do this:

- a. Open the situation
- b. Click the **Advanced** tab
- c. Select **Display Item** and the Filename attribute.

The following table lists the pre-configured actions for this monitoring resource:

Table 8. Events formats table

Severity	Trigger When	Default Actions
Critical	Message file megabytes used are greater or equal to 90% of the total allocated size	Send event

Table 8. Events formats table (continued)

Severity	Trigger When	Default Actions
Warning	Message file megabytes used are greater or equal to 50% of the total allocated size	Send event
Informational	Message file megabytes used are less than 50% of the total allocated size	Send event

Integrating with Tivoli Enterprise Portal using a file data provider

This section describes the integration of Tivoli Workload Scheduler and dynamic workload broker with Tivoli Enterprise Portal, a component of Tivoli Monitoring 6.1.

For information about the integration of Tivoli Workload Scheduler with Tivoli Enterprise Portal, see “Integrating Tivoli Workload Scheduler with Tivoli Enterprise Portal.”

For information about the integration of dynamic workload broker with Tivoli Enterprise Portal, see “Integrating dynamic workload broker with Tivoli Enterprise Portal” on page 19.

Integrating Tivoli Workload Scheduler with Tivoli Enterprise Portal

This section describes the integration of Tivoli Workload Scheduler with Tivoli Enterprise Portal, a component of Tivoli Monitoring 6.1.

It is divided into the following subsections:

- “Architecture of the integration”
- “Configuring the integration with Tivoli Enterprise Portal” on page 10
- “Displaying events on the Tivoli Enterprise Portal desktop” on page 15
- “Events displayed by Tivoli Enterprise Portal” on page 15
- “Collecting and reloading monitoring data” on page 19

Architecture of the integration

Tivoli Monitoring monitors and manages system and network applications on a variety of platforms and keeps track of the availability and performance of all parts of your enterprise. Tivoli Monitoring provides reports you can use to track trends and troubleshoot problems.

Tivoli Workload Scheduler sends events to Tivoli Monitoring 6.1 using the File Data Provider Feature of the Tivoli Universal Agent.

Tivoli Workload Scheduler creates and writes events for all the monitored objects (jobs and job streams) into a predefined event file that is accessed by the Tivoli Universal Agent at a set interval. The Universal Agent reads the events from the event file and interprets the data using the information contained in a metafile which is provided with Tivoli Workload Scheduler and previously loaded on the Tivoli Universal Agent by a script called **TEPConfig**. The Tivoli Universal Agent stores the event information in tables that are consolidated by the Tivoli Enterprise

Monitoring Server. The collected data is then retrieved and displayed by a component called Tivoli Enterprise Portal.

The following components of Tivoli Monitoring 6.1 with Fix Pack 5 must be installed on one of the allowed platforms to allow the integration with Tivoli Workload Scheduler:

- A Tivoli Enterprise Monitoring Server (referred to as the monitoring server), which acts as a collection and control point for the events received from the agents.
- A Tivoli Enterprise Portal Server (referred to as the portal server) placed between the client and the monitoring server, that enables retrieval, manipulation, and analysis of data from the agents.
- A Tivoli Enterprise Portal client with a Java-based user interface for viewing and monitoring your enterprise. Tivoli Enterprise Portal offers two modes of operation: desktop and browser.
- A Tivoli Universal Agent installed on the systems that you want to monitor. In your Tivoli Workload Scheduler environment, this component must be installed on the master domain manager system. This agent collects and distributes data to the monitoring server.

The following figure describes the architecture of the integration:

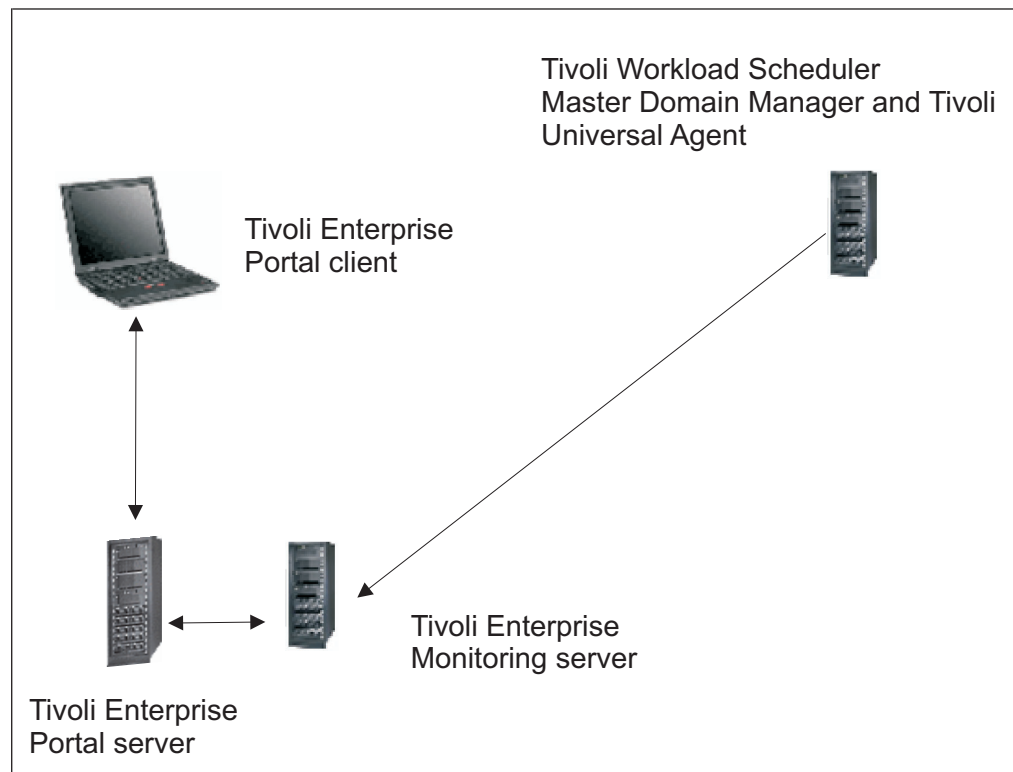


Figure 5. IBM Tivoli Monitoring architecture

Configuring the integration with Tivoli Enterprise Portal

This section describes the steps you must perform to enable monitoring by the Tivoli Enterprise Portal.

To enable monitoring, run the configuration script called **TEPconfig.sh** for UNIX environments or **TEPconfig.cmd** for Windows environments on the computer where the Tivoli Workload Scheduler master domain manager and the Universal Agent component of IBM Tivoli Monitoring 6.1 with fix pack 5 are installed.

1. Launch the script as follows:

```
TEPConfig -ua_home_dir <ua_home_dir> [eventfile_path <eventfile_path>]  
[-metafile_path <metafile_path>] [-APPL_NAME <APPL_NAME>] [-TTL <TTL>]  
[-SAMPLE_FACTOR <SAMPLE_FACTOR>] [-UA]
```

where:

<ua_home_dir>

Is the home directory path of the Tivoli Universal Agent installation. This parameter is mandatory.

<eventfile_path>

Is the fully qualified path of the event log file. The default value is <TWS_HOME_DIR>/event.log. If the path contains blanks, you must enclose it between double and single quotes as in the following example:

```
-eventfile_path="'c:\Program Files\IBM\TWS\TEEvent.log'
```

<metafile_path>

Is the fully qualified path to the sample metafile. The default value is <TWS_HOME_DIR>/TEP/TWS84metafile_Sample.mdl. If the path contains blanks, you must enclose it between double quotes as in the following example:

```
-metafile_path="c:\Program Files\IBM\TWS\TEP\TWS84metafile_Sample.mdl"
```

<APPL_NAME>

Is the application name that will be stored in the APPL statement of the metafile. The default value is **dtWS**. The value specified must be between 3 and 20 alphanumeric characters including the underscore (**_**), dash (**-**), and the asterisk (*****). For details about the allowed values for this parameter refer to the *Tivoli Universal Agent User's Guide*.

<TTL>

Is the amount of time in seconds (Time To Live) that the monitored data will be kept by the Tivoli Enterprise Portal. This value is set in the TTL keyword of the NAME statement in the metafile. The default value is 86400 seconds. For details about the NAME statement refer to the *Tivoli Universal Agent User's Guide*.

<SAMPLE_FACTOR>

Is the value to set for the KUMP_DP_SAMPLE_FACTOR parameter of the Universal Agent initialization file (KUMENV on Windows platforms or um.ini on UNIX platforms). The default value is 5760. It is used to calculate the sampling frequency used by the Universal Agent to access the event file. This frequency is obtained by dividing the TTL value (default of 86400) by the KUMP_DP_SAMPLE_FACTOR. If you keep the default values for both the TTL and SAMPLE_FACTOR parameters, the event file is accessed every 15 seconds. For details about the allowed values for this parameter refer to the *Tivoli Universal Agent User's Guide*.

-UA

Use this parameter if you want to modify Universal Agent related parameters only (**APPL_NAME**, **TTL** or **SAMPLE_FACTOR**) without changing any of the existing settings in the **BmEvents.conf** configuration file (if present). When you specify this parameter, if a

configuration file already exists, it is not affected. If it does not exist it is not created. If this parameter is specified the first time you run the **TEPConfig** script, you must ensure that a copy of the **BmEvents.conf** file is present in the Tivoli Workload Scheduler home directory, by manually copying it to the correct path.

2. Recycle the Tivoli Workload Scheduler engine. You must always perform this step in the following cases:
 - After you run **TEPConfig** for the first time.
 - After any run of **TEPConfig** in which you add or modify `ua_home_dir`, `eventfile_path` or `metafile_path` parameter values.

Note: If you are using special characters in job and job stream names, you need to add the following keyword to the universal agent initialization file (`KUMENV` on Windows or `um.ini` on UNIX platforms) to ensure correct interpretation of these characters by Tivoli Enterprise Portal:

```
KUMP_DEFAULT_CODEPAGE=UTF-8
```

Restart the universal agent after this change.

The script performs the following steps:

1. Checks if a **BmEvents.conf** file already exists in the Tivoli Workload Scheduler home directory. If the file exists, it is renamed to **BmEvents.conf.prev** to maintain any existing customizations. A new copy of **BmEvents.conf** is made in the home directory which includes the existing customizations and the new values needed for the integration with Tivoli Enterprise Portal. If the **BmEvents.conf** file is not found in the Tivoli Workload Scheduler home directory, the `<TWS_HOME_DIR>\config\BmEvents.conf` (on Windows environments) or the `<TWS_HOME_DIR>/OV/BmEvents.conf` (on UNIX environments) is copied to the Tivoli Workload Scheduler home directory.
2. Adds a list of events to the configuration file in the **EVENT** parameter
3. Customizes the sample metafile by setting the values for the `APPL_NAME`, `eventfile_path`, and `TTL` parameters, if different from the default values.
4. Performs the import or refresh of the customized metafile **TWS84metafile.mdl**. The Universal Agent uses this metafile to interpret the information stored in the event log and passes this monitoring data to the Tivoli Enterprise Monitoring Server.
5. If the import or refresh of the metafile was successful, an automatic restart of the Universal Agent is performed. If any warning or error messages were issued by the Universal Agent during the process, a restart is not performed. In this case, you must manually restart the Universal Agent. A restart is required in the following cases:
 - It is the first time you run the **TEPConfig** script, or
 - Each time you change the value of the `SAMPLE_FACTOR` parameter.

For a detailed description of the fields modified by the customization script, refer to “The **BmEvents.conf** file” on page 13

When the Tivoli Enterprise Portal integration is enabled, for each production plan the Tivoli Workload Scheduler master domain manager logs events related to jobs and job streams in the specified event file. The Tivoli Universal Agent retrieves the logged events from the event file and parses them using the installed metafile. The monitored data is then passed by the Tivoli Universal Agent to the Tivoli Enterprise Monitoring Server and to the Tivoli Enterprise Portal Server. The Tivoli Enterprise Portal Client displays the collected data.

The BmEvents.conf file:

The **TEPConfig** script sets up the file **BmEvents.conf** in the *TWSHome* directory. This configuration file determines what information the production processes (batchman and mailman) write in the *TWSHome/log_source_file* file, by default this file is the *event.log* file, and how this information is written.

You can change the name of the log file as follows:

- Run the **TEPConfig** script specifying the `<eventfile_path>` parameter. The value you specify changes both the *BmEvents.conf* and the metafile information so that both the Universal Agent and Tivoli Workload Scheduler are synchronized. This is the preferred method.
- Modify the **FILE** field in the *BmEvents.conf* file and restart the Tivoli Workload Scheduler processes. In this case, you also have to manually change the metafile information and perform a refresh of the Universal Agent configuration to synchronize the Universal Agent and IBM Tivoli Workload Scheduler.

In the *BmEvents.conf* file the `#` sign represents a comment. Remove the `#` sign to uncomment a line.

The options you can set in the *BmEvents.conf* file are described below:

OPTIONS=MASTER|OFF

Determines for which workstations the gathered events are reported.
Possible values are:

MASTER

All job scheduling events gathered by that workstation are reported. If that workstation is the master domain manager or the backup master domain manager, with Full Status option switched on, then all scheduling events for all workstations are reported.

OFF The job scheduling events are reported only if they relate to the workstation where the file is configured.

If commented, it defaults to MASTER on the master domain manager workstation, and to OFF on a workstation other than the master domain manager.

LOGGING=ALL|KEY

Disables or enables the key flag filter mechanism. Possible values are:

ALL If set to this value, all events from all jobs and job streams are logged.

KEY If set to this value, the event logging is enabled only for those jobs and job streams that are marked as key. The key flag is used to identify the most critical jobs or job streams. To set it in the job or job stream properties use:

- The keywords **KEYSCHED** (for job streams) and **KEYJOB** (for jobs) from the Tivoli Workload Scheduler command line interface.
- The job Monitored check box and job stream Monitored check box from the Dynamic Workload Console.

The **TEPConfig** script sets the value of this parameter to **KEY**.

SYMEVNTS=YES|NO

Used in conjunction with **LOGGING=KEY**. It determines whether events concerning jobs and job streams are to be reported immediately after a plan creation.

YES If set to this value, it tells the production process, **batchman**, to report the jobs and job streams status events immediately after having generated the new production day plan. This key is valid only if **LOGGING=KEY**. For the integration with Tivoli Enterprise Portal, this value will cause a bulk discovery to be automatically performed after each new plan creation.

NO If set to this value, no report is given.

The default value is NO.

CHSCHEM=HIGH|LOW

Indicates which events are to be sent during the job stream lifetime.

During the lifetime of a job stream its status can change several times depending on the status of the jobs it contains. By using the **CHSCHEM** option you choose how the job stream status change is reported.

HIGH If you set tis value, during the job stream lifetime an event is sent any time the status of the job stream changes. Because the intermediate status of the job stream can change several times, several events can be sent, each reporting a specific status change. For example, a job stream may go into the READY state several times during its running because its status is related to the status of the jobs it contains. Each time the job stream goes into the READY state, event 153 is sent.

LOW If you set this value, during the job stream lifetime until the final status is reached, only the initial job stream state transaction is tracked. In this way the network traffic of events reporting job stream status changes is heavily reduced.

When the **CHSCHEM** value is set to LOWthese are the events that are sent only the first time during the job stream life time:

Table 9. CHSCHEM filtered events

Event number	Event Class	Description
153	TWS_Schedule_Started	Job stream started
156	TWS_Schedule_Submit	Job stream submitted
158	TWS_Schedule_Ready	Job stream ready
159	TWS_Schedule_Hold	Job stream hold
160	TWS_Schedule_Extern	Job stream external
162	TWS_Schedule	Job stream properties changed

For final status of a job stream, regardless of the value set for **CHSCHEM**, all events reporting the final status of the job stream are reported, even if the job stream has more than one final status. For example, if a job contained in the job stream completes with an ABEND state, event 151 is sent (Job stream abended). If that job is then rerun and completes successfully, the job stream completes with a SUCC state and event 154 is sent (Job stream completed).

The default value for **CHSCHEM** is HIGH.

EVENT=*n* [*n* ...]

Identifies which events to report in the `log_source_file`. Event numbers must be separated by at least one space. The events added by the configuration script are the following:

```
101 102 103 104 105 106 107 108 109 110 111 112 113 114 116 117 118 120
121 122 151 152 153 154 155 156 157 158 159 160 161 163 164 165 204
```

If the **EVENT** parameter is manually commented out, the events read by the Tivoli Universal Agent in the event file are:

```
101 102 105 151 152 155 204
```

FILE=*filename*

This is the path and file name of an ASCII log file where job scheduling events are written. This parameter is set by the configuration script. This file is truncated whenever the **batchman** and **mailman** processes are restarted, for example at the end of each production day.

Displaying events on the Tivoli Enterprise Portal desktop

To display Tivoli Workload Scheduler events from your Tivoli Enterprise Portal desktop, open the default workspace provided for the TWSd application. This default view contains two tables that show Tivoli Workload Scheduler jobs and job streams .

Optionally, you can install the customized workspaces and situations by using the programs included with Tivoli Workload Scheduler.

These programs are found in the **CD_ROOT** directory of your installation CD. To use the customized solution, do the following:

1.

- 1 On the computer where your Tivoli Enterprise Portal Server is installed, use
1 one of the following installation programs, depending on the operating system:
1
 - 1 • `seupaix.bin`
 - 1 • `setupwin32.exe`
 - 1 • `setuplinux390.bin`
 - 1 • `setupLinux.bin`
 - 1 • `setupSolaris.bin`

2. Launch the installation program and proceed with the installation. In the directory window, enter the path where you installed Tivoli Monitoring. The second box contains the location from where the installation was launched.
3. On the application selection screen, select **Tivoli Workload Scheduler Customized Solution** and click **Next**. The final verification screen opens.
4. To start the installation, click **Next**.
5. Log on to the TEP client to view the pre-defined workspaces, situations and queries .

Events displayed by Tivoli Enterprise Portal

Table 10 on page 16 lists the events logged for the integration with Tivoli Enterprise Portal.

Table 10. Tivoli Workload Scheduler engine event formats for Tivoli Enterprise Portal

Event	Number	Description
mstJobAbend	101	Job abended
mstJobFailed	102	Job failed
mstJobLaunch	103	Job launched
mstJobDone	104	Job done
mstJobUntil	105	Job suspended until expired
mstJobSubmit	106	Job submitted
mstJobCancel	107	Job cancelled
mstJobReady	108	Job in READY status
mstJobHold	109	Job in HOLD status
mstJobRestart	110	Job restarted
mstJobCant	111	Job Failed
mstJobSuccp	112	Job Successful pending
mstJobExtrn	113	Job extern
mstJobIntro	114	Job in INTRO status
mstJobWait	116	Job in WAIT status
mstJobWaitd	117	Job in wait deferred status
mstJobSched	118	Job in scheduled status
mstJobLate	120	Job is late
mstJobUntilCont	121	Job UNTIL time expired with Continue option
mstJobUntilCanc	122	Job UNTIL time expired with Cancel option
mstSchedAbend	151	Job Stream abended
mstSchedStuck	152	Job stream is stuck
mstSchedStart	153	Job stream started
mstSchedDone	154	Job stream done
mstSchedUntil	155	Job Stream suspended, until time expired
mstSchedSubmit	156	Job stream submitted
mstSchedCancel	157	Job Stream cancelled
mstSchedReady	158	Job stream in READY status
mstSchedHold	159	Job stream in HOLD status
mstSchedExtrn	160	Job stream extern

Table 10. Tivoli Workload Scheduler engine event formats for Tivoli Enterprise Portal (continued)

Event	Number	Description
mstSchedCnpend	161	Job Stream in CANCEL Pending status
mstSchedLate	163	Job Stream is late
mstSchedUntilCont	164	Job Stream Until time expired with continue option
mstSchedUntilCanc	165	Job Stream until time expired with cancel option
mstJobRecovPrompt	204	Job Recovery prompt issued
mstJobBound	308	For shadow jobs: the shadow job matched a remote job instance in the remote plan. For TWS for z/OS® agents: the job is on the JES queue.

Positional event variables:

This section defines the positional event variables.

Table 11. Positional variables for events 101-118,120-122, 204 (job events)

Variable	Description
1	Event number
2	Job stream cpu
3	Job Stream id
4	Job name
5	Job CPU
6	Job number
7	Job status
8	Real name (different from job name only for MPE jobs)
9	Job user
10	Jcl name (script name or command name)
11	Every time
12	Recovery status
13	Time stamp (yyyymmddhhmm0000)
14	Message number (not equal to zero only for job recovery prompts)
15	Eventual text message (delimited by '\t')
16	Record number
17	Key flag

Table 11. Positional variables for events 101-118,120-122, 204 (job events) (continued)

Variable	Description
18	Effective start time
19	Estimated start time
20	Estimated duration
21	Deadline time (epoch)
22	Return code
23	Original schedule name (schedule name for schedules not (yet) carried forward)
24	Head job record number (different from record number for rerun/every jobs)
25	Job stream name
26	Job stream scheduled time (yyyymmddhhmm00)
27	Event Version
28	Resource Name
29	Resource Type
30	Job external status
31	Job ID
32	Name space
33	Submitter
34	Job end time
35	Job submit time
36	Cpu usage
37	Memory usage
38	Operation number
39	Job error code
40	Critical job identifier

Table 12. Positional variables for events 151-161, 163-165 (job stream events)

Variable	Description
1	Event number
2	Job stream CPU
3	Job stream ID
4	Job stream status
5	Record number
6	Key flag
7	Original job stream name (job stream name for job streams not (yet) carried forward)
8	Time stamp (yyyymmddhhmm0000)
9	Job stream name
10	Job stream scheduled time (yyyymmddhhmm00)
11	Event version

Collecting and reloading monitoring data

When the integration is enabled, the Tivoli Workload Scheduler engine starts to log monitoring events after each new plan production. To reload all monitoring data, or to obtain the initial status of your monitored objects, use the conman **bulk_discovery** command. This command checks the current status of all the monitored objects (jobs and job streams) within the Symphony™ file and writes the corresponding events in the event.log file for retrieval by the Tivoli Enterprise Portal.

Note:

1. If a JnextPlan is requested while a bulk_discovery is in progress, some events related to jobs and job streams may no longer be visible from the Tivoli Enterprise Portal console views when the new plan is built. Inconsistencies might occur because the JnextPlan process deletes and reallocates the event file at the end of its activity. If the Universal Agent has not read all records of the event file before the file is reallocated, it is not aware of any missing job or jobstream status until a new change of status occurs. To ensure consistency, issue a bulk_discovery command at the end of the JnextPlan, or set **SYMEVNTS=YES** in the **BmEvents.conf** file so that a bulk discovery is performed automatically after every JnextPlan activity.
2. If you recycle the Tivoli Universal Agent, events are lost. If the Universal Agent is stopped for any reason, also as a result of a TCP/IP connection failure, events are lost unless you have saved them using the Universal Agent history function. In these cases, issue a bulk discovery to refresh the status of all monitored objects.

For details about the syntax and output of the bulk_discovery command, refer to *IBM Tivoli Workload Scheduler: User's Guide and Reference*.

Integrating dynamic workload broker with Tivoli Enterprise Portal

This topic describes the integration of dynamic workload broker with Tivoli Enterprise Portal version 6.2.1, a component of IBM Tivoli Monitoring, and the steps required to complete the integration. It includes the following sections:

- "Tivoli Monitoring components" on page 20
- "Integration overview" on page 20
- "Before you integrate" on page 21
- "Configuring the integration with Tivoli Enterprise Portal" on page 21
- "The TEPListener.properties file" on page 23
- "TEPBulkDiscovery command - Performing a bulk discovery" on page 24

The integration with Tivoli Enterprise Portal provides extensive monitoring capabilities that are important in an environment where jobs are automatically distributed among available resources. Using this integration, events regarding the dynamic workload broker processes can be viewed using the Tivoli Enterprise Portal. To each status of a broker job is associated an event that you can monitor using the Tivoli Enterprise Console®. This integration provides automated problem diagnosis and resolution to improve system performance and reduce support costs. Through this integration, events regarding the broker jobs are written in the Data Warehouse database and you can use them to produce reports and statistics.

Tivoli Monitoring components

The Tivoli Monitoring components involved in the integration are:

Tivoli Enterprise Monitoring Server

Is the central repository of data that comes from the Tivoli Enterprise Monitoring Agents. It acts as a collection and control point for alerts from agents and collects performance and availability data from the agent. It also stores the definitions for conditions that indicate possible problems with monitored resources.

Tivoli Enterprise Portal Server

Is a repository for all user data, such as user IDs and the settings that control user access to collected data. It controls the data that a user can see and the way it is displayed.

Tivoli Enterprise Portal

Is the presentation layer for displaying the monitoring data. It offers a consolidated view of the whole IT environment.

Tivoli Universal Agent

Collects data from the managed systems and transfers it to the Tivoli Enterprise Monitoring Server to which it is connected. The IBM Tivoli Universal Agent retrieves data from interfaces called data providers. A log file adapter collects information from dynamic workload broker logs. You can create data definitions that describe the source and structure of the data supplied by the data providers. You can store the data definitions in metafiles.

Integration overview

The integration with Tivoli Enterprise Portal is installed automatically when you install the dynamic workload broker server and does not require user input during the installation. Afterwards, you can enable and configure the integration with the **TEPConfig** command. For more information, see “Configuring the integration with Tivoli Enterprise Portal” on page 21.

When the integration is enabled, a dynamic workload broker process called **TEPListener** waits for occurrences of events that are to be tracked (jobs with statuses that you have chosen to monitor) and adds them to the log file, **TEPEVENTyyyymmddHHMM.log**. The Universal Agent regularly parses this log file and finds any newly appended lines. When a new line is found, it is analyzed against the definitions included in the agent metafile. If the line matches one of the definitions, parsed data is sent to the Tivoli Enterprise Portal server and is graphically presented in the Tivoli Enterprise Portal.

The Tivoli Enterprise Portal navigation panel lists an application name consisting of the dynamic workload broker server host name, an application name specified in the agent metafile, followed by a two-digit suffix representing the version and modification level.

Log and trace information for this integration is stored in the Websphere Application Server logs and traces in the following path: *WAS_profile_root* /AppSrv01/logs/ server1.

Before you integrate

Before running the command to enable and configure the integration, perform the following:

- Ensure that the Tivoli Monitoring Universal Agent is installed on the computer where the dynamic workload broker server is installed. Ensure that you know the installation path of the agent.
- Choose the path where you want dynamic workload broker to create the log file. If you decide to override the default directory, ensure that the directory exists before you launch the command. No special access permissions are required.
- Prepare a list of the job states that you want to be monitored by Tivoli Enterprise Portal.

Configuring the integration with Tivoli Enterprise Portal

Use the **TEPConfig** command to do the following:

- Configure dynamic workload broker so that it logs required events to the file that will be read by the Tivoli Monitoring Universal Agent.
- Configure the Tivoli Monitoring Universal Agent so that it is able to understand the content of the log file.
- Configure the views in the Tivoli Enterprise Portal so that situations that originate from dynamic workload broker can be displayed.

The settings you define with the **TEPConfig** command are stored in the `TEPListener.properties` file. After running the **TEPConfig** command, or changing the `TEPListener.properties` file, you must stop and restart the Websphere Application Server to make effective the changes to the file.

Syntax

TEPConfig ?

```
TEPConfig {-UAIInstDir Universal Agent installation directory [-eventFilePathName event log file path] [-metafileName metafile name ] [-UAApplName application name for TEP] [-UA_TTL monitor time to live]} | {[-eventFileSize event file maximum size] [-events event...]}
```

Description

This command configures the parameters of the integration with Tivoli Enterprise Portal. When you run the command, the following line is written in the `JobDispatcherConfig.properties` file, enabling the integration with Tivoli Enterprise Portal:

```
JobStatusChangeListener.TEP=com.ibm.scheduling.jobdispatcher.  
jobstatuslistener.TEPJobStatusChangeListener
```

The command is stored on dynamic workload broker servers in the following location:

```
installation_directory\bin
```

Options

? Displays help information.

-UAIInstDir *Universal Agent installation directory*
Specifies the home directory of the IBM Tivoli Monitoring installation for the

Tivoli Universal Agent. This option is mandatory when you are modifying parameters stored in the metafile. For example, if you are using a Windows system and have installed IBM Tivoli Monitoring in the C:\IBM\ITM directory, and the Tivoli Universal Agent in C:\IBM\ITM\TMAITM6, the **UAInstDir** option must be set to C:\IBM\ITM. Do not use paths containing spaces because they are not supported.

-eventFilePathName *event log file path*

Specifies the path for the event log files stored in the metafile. The default value is *installation_directory/logs*. When using this option, you must also specify the **-UAInstDir** option. Do not use paths containing spaces because they are not supported.

-metafileName *metafile name*

Specifies the name of the metafile containing the monitoring data syntax. This option is optional. The default value is *installation_directory/TEP/TDWBMeta.mdl*. When using this option, you must also specify the **-UAInstDir** option. Do not use paths containing spaces because they are not supported.

-UAAppName *application name for TEP*

Optional. Specifies the name of the application to be monitored that can be customized in the metafile syntax for the APPL statement stored in the metafile. The default value is **TDWB**. The application name must be from 3 to 20 characters. When using this option, you must also specify the **-UAInstDir** option. For more information about the APPL statement, refer to the *IBM Tivoli Universal Agent User's Guide*.

-UA_TTL *monitor time to live*

Optional. Specifies the number of seconds that can be specified in the TTL (time to live) keyword of the NAME statement stored in the metafile. The default value is 86400 seconds (1 day). This value specifies the time interval that the events are maintained in the Tivoli Enterprise Portal database before they are deleted. When using this option, you must also specify the **-UAInstDir** option.

-eventFileSize *event file maximum size*

Specifies the maximum size that an event log can reach before switching to the alternative one. The default value is 20 MB.

-events *event...*

Specifies a list of the job statuses that are to be added to the list of events to monitor. Supported values are as follows:

-
- CANCEL
- CANCEL_ALLOC
- EXEC
- FAILED
- PENDING_CANCEL
- RES_ALLOC_RECEIVED
- RES_ALLOC_FAILED
- RES_REALLOC
- RES_REALLOC_FAILED
- SUBMITTED
- SUBMITTED_TO_ENDPOINT
- SUCC

- UNKNOWN
- WAIT_FOR_RES

By default, jobs in the following statuses are monitored:

- CANCEL
- FAILED
- NOT_EXECUTED
- RES_ALLOC_FAILED
- RES_REALLOC

Note: This option adds specified statuses to those that are already being monitored. It does not replace the current list of monitored statuses. If you want to remove a job status from the list, edit the `TEPConfig.properties` file.

Authorization

None

Return Values

The **TEPConfig** command returns one of the following values:

- 0** Indicates that **TEPConfig** completed successfully.
- < > 0** Indicates that **TEPConfig** failed.

Examples

1. To modify the event file size to 40 MB and to log information about jobs in waiting for resources and submitted status, type the following command:

```
TEPConfig -UAInstDir /home/IBM/ITM -eventFileSize 40 - events \  
WAIT_FOR_RES SUBMITTED
```

You must specify the Tivoli Universal Agent installation directory because the parameters you are modifying are stored in the metafile, which must be replaced.

See Also

None.

The TEPListener.properties file

You can use the `TEPListener.properties` file to configure the parameters of the integration with dynamic workload broker. The `TEPListener.properties` file is installed in the following path:

installation_directory/config

The following parameters are available in the `TEPListener.properties` file:

MAXEVTSIZE

Specifies the maximum size that an event log can reach before switching to the alternative one. The default value is 20 MB.

EVENTFILEPATH

Specifies the path for the event log files. The default value is *installation_directory/logs*.

In addition, there is a parameter for each job event that is monitored which specifies if jobs with that status are to be monitored. Supported values are **yes** and **no**. The following are the job events that can be monitored:

Table 13. Job events

Parameter name	Default setting
CANCEL	yes
CANCEL_ALLOC	no
EXEC	no
FAILED	yes
NOT_EXECUTED	yes
PENDING_CANCEL	no
RES_ALLOC_FAILED	yes
RES_ALLOC_RECEIVED	no
RES_REALLOC	yes
RES_REALLOC_FAILED	no
SUBMITTED	no
SUBMITTED_TO_ENDPOINT	no
SUCC	no
UNKNOWN	no
WAIT_FOR_RES	no

To make effective the changes to the file, you must stop and restart the Websphere Application Server.

TEPBulkDiscovery command - Performing a bulk discovery

A bulk discovery is performed to load current job data from the Job Repository. The bulk discovery uploads job information stored in the Job repository database to the Tivoli Enterprise Portal Server. The query for the bulk discovery is performed on jobs that have finished running, regardless of their final status.

To perform a bulk discovery, you can use the following command:

Syntax

TEPBulkDiscovery ?

```
TEPBulkDiscovery [-usr user_name -pwd password] { [-submitter submitter] [-name job_definition_name] [-alias job_alias] [-sdf submit_date_from] [-sdt submit_date_to] [-jsdf job_start_date_from] [-jsdt job_start_date_to] [-jedf job_end_date_from] [-jedt job_end_date_to]} [-configFile configuration_file]
```

Description

This command performs advanced queries on submitted jobs based on the following attributes:

- Job status
- Name of the user who submitted the job

- Job name
- Job alias
- Job submission date
- Job start date
- Job completion date

You can also use this command to retrieve the job ID generated at submission time, which is required when running the **jobstatus**, **jobdetails**, and **jobcancel** commands. To retrieve the job ID, specify the **-name** option.

Options

? Displays help information.

-usr *user name*

Specifies the user name for a user who is authorized to perform operations on the command line. This option is required when security is enabled and the user name is not defined in the CLIconfig.properties configuration file.

-pwd *password*

Specifies the password for a user authorized to perform operations on the command line. This option is required when security is enabled and the password is not defined in the CLIconfig.properties configuration file.

-submitter *submitter*

Specifies the name of the user who submitted the job.

-name *job_definition_name*

Specifies the job name. This option returns the unique job ID, which can be used for retrieving information about and canceling jobs. This option supports the asterisk (*) wildcard character as described below:

As a single parameter

It must be enclosed in inverted commas, for example

```
C:\Program Files\TDWB\bin>TEPBulkDiscovery -name "*"
```

This command returns a list of all the submitted jobs.

To complete a job name

It does not require inverted commas, for example

```
C:\Program Files\TDWB\bin>TEPBulkDiscovery -name batchsub*
```

This command returns a list of all the submitted jobs that start with the **batchsub** prefix.

-alias *job_alias*

Specifies the job alias. The job alias is generated at submission time using the **-alias** option.

-sdf *submit_date_from*

Specifies a time range starting from the date when the job was submitted. The query is performed starting from the date you specified to the current date, unless the **-sdt** option is specified. Use both the **-sdf** and **-sdt** options to define a specific time range. Specify the date in the dd/MM/yyyy-hh:mm:ss format.

-sdt *submit_date_to*

Specifies a time range starting from the date when the job was submitted. The query is performed starting from the date you specified to the date when the dynamic workload broker database was populated, unless the **-sdf** option is

specified. Use both the **-sdf** and **-sdt** options to define a specific time range. Specify the date in the dd/MM/yyyy-hh:mm:ss format.

-jsdf *job_start_date_from*

Specifies a time range starting from the date when the job started. The query is performed starting from the date you specified to the current date, unless the **-jsdt** option is specified. Use both the **-jsdf** and **-jsdt** options to define a specific time range. Specify the date in the dd/mm/yyyy-hh:mm:ss format.

-jsdt *job_start_date_to*

Specifies a time range starting from the date when the job started. The query is performed starting from the date you specified to the date when the dynamic workload broker database was populated, unless the **-jsdf** option is specified. Use both the **-jsdf** and **-jsdt** options to define a specific time range. Specify the date in the dd/MM/yyyy-hh:mm:ss format.

-jedf *job_end_date_from*

Specifies a time range starting from the date when the job completed. The query is performed starting from the date you specified to the current date, unless the **-jedt** option is specified. Use both the **-jedf** and **-jedt** options to define a specific time range. Specify the date in the dd/mm/yyyy-hh:mm:ss format.

-jedt *job_end_date_to*

Specifies a time range starting from the date when the job completed. The query is performed starting from the date you specified to the date when the dynamic workload broker database was populated, unless the **-jedf** option is specified. Use both the **-jedf** and **-jedt** options to define a specific time range. Specify the date in the dd/mm/yyyy-hh:mm:ss format.

-configFile *configuration_file*

Optional. Specifies the name and path of a custom configuration file. If this option is not specified, the default configuration file is used.

Authorization

The user name and password for the command are defined in the CLIConfig.properties file. To override the setting defined in this file, you can enter the user name and password when typing the command.

Return Values

The **TEPBulkDiscovery** command returns one of the following values:

- 0** Indicates that **TEPBulkDiscovery** completed successfully.
- < > 0** Indicates that **TEPBulkDiscovery** failed.

Examples

1. To retrieve the job ID for a job named CLIJSB11, type the following command:
TEPBulkDiscovery -usr john -pwd BCA12EDF -name CLIJSB11
2. To retrieve all jobs submitted by test_user in submitted, resource allocation failed, and canceled state, type the following command:
TEPBulkDiscovery -status 1,3,44 -submitter test_user

Scenario: integrating with Tivoli Enterprise Portal

This scenario describes how Tivoli Workload Scheduler and the Dynamic Workload Console integrate with IBM Tivoli Monitoring, the IBM solution for controlling and managing the performance and availability of distributed operating systems and applications, and its Java-based user interface, the Tivoli Enterprise Portal. You can perform these operations from a centralized point instead of having to combine and integrate data from different products.

This integration allows you to control from the Tivoli Enterprise Portal, jobs and job streams that are critical to run your business, and to quickly and easily manage them on the Dynamic Workload Console.

With just one click, you launch the Dynamic Workload Console from the Tivoli Enterprise Portal interface and access specific job or job stream details without having to create customized queries.

Starting from Tivoli Workload Scheduler V8.6, the Launching in context with the Dynamic Workload Console can be useful especially to rapidly view more details about the new job types. From the Tivoli Enterprise Portal, in fact, you can directly open the job log to view the JSDL definition of these JSDL-based job types.

The integration also allows you to monitor the state and health of the Tivoli Workload Scheduler workstations that are critical in your environment so that, when unavailability or malfunctioning impacts the job scheduling capability, you are alerted on the Tivoli Enterprise Portal interface.

In this scenario you use:

IBM Tivoli Monitoring

To control the status and the health of your core business environment. You also outsource to a third-party company the monitoring of your environment from the Tivoli Enterprise Portal user interface.

Tivoli Workload Scheduler

To run, within a selected time frame, a daily backup of the IBM Tivoli Monitoring data and configuration to ensure the monitoring reliability and availability required to satisfy the SLAs agreed with the third party company. This daily backup ensures your core business runs successfully.

Dynamic Workload Console

To access specific job or job stream details.

Scenario goal

Your goal is to ensure that the activities critical for your core business run successfully. When a critical job, used to run the daily backup, stops running unexpectedly, you interact with that job by launching Dynamic Workload Console from the Tivoli Enterprise Portal interface. You can restart processing to ensure that the backup activity is completed successfully within the selected time frame.

Required roles and skills

The role and the required skill level of the user running the scenario are the following:

For the setup phase:

For IBM Tivoli Monitoring:

Either a user in the ADMINISTRATOR user group or a user in the OPERATOR user group, with the following permissions:

- List and modify permission on Managed System List.
- Modify permission on Situation.
- Modify permission on Launch Application.

For Tivoli Workload Scheduler:

Either the Tivoli Workload Scheduler administrator or a Tivoli Workload Scheduler user allowed in the `security.conf` file to run the listed steps. For more information about how to set the authorization in the `security.conf` file, refer to *IBM Tivoli Workload Scheduler: Administration Guide*.

For the using phase:**For IBM Tivoli Monitoring:**

Either a user in the ADMINISTRATOR user group or a user in the OPERATOR user group, with the following permissions:

- Launch permission on Launch Application.
- View permission on Event.
- Logon Permitted on User Administration.

For Dynamic Workload Console:

A user belonging to a group that has assigned either the *TWSWEBUIAdministrator* or *TWSWEBUIOperator* role.

For more information, refer to the *IBM Tivoli Workload Scheduler: Administration Guide*.

In this scenario, we assume that the Single Sign-On method has been configured between Dynamic Workload Console and the Tivoli Workload Scheduler master domain manager and that the user who logs in to the Tivoli Workload Scheduler master domain manager through the engine connection is allowed to run the steps specified in “Running the scenario” on page 34.

Software requirements

Install the following software before starting the scenario:

- Tivoli Workload Scheduler 8.4 Fix Pack 01 or later (master domain manager, agents, Dynamic Workload Console)
- The following components of IBM Tivoli Monitoring version 6.1 Fix Pack 5 or later:

Tivoli Universal Agent

It is a generic agent of IBM Tivoli Monitoring that can be configured to monitor any data collected from a data provider.

Tivoli Enterprise Monitoring Server

It is the core component of the IBM Tivoli Monitoring product. It collects and controls the events received from the agents. It uses a proprietary database to store the monitoring information.

Tivoli Enterprise Portal Server

It enables retrieval, manipulation, and analysis of data from the agents. It uses an external relational database, such as DB2®, to store the data.

Tivoli Enterprise Portal

It is the Java-based Tivoli Enterprise Portal Server user interface for viewing and monitoring your enterprise.

Note:

- A problem in IBM Tivoli Monitoring prevents you from using Internet Explorer for the Dynamic Workload Console launch in context from the Tivoli Enterprise Portal user interface. The limitation is documented in PMR number 02306,758,758.
- All .TAR and BAROC files for the Tivoli Enterprise Portal integration are located on the installation DVD in the **integrations\TEP** directory.

The following figure shows the information flow as a result of the integration. In this scenario, we assume that each block represents a different workstation where you install and set up the indicated components. You can decide where to install each of these blocks in your environment according to your environment requirements and workstation resources. For information about system requirements for each component, refer to the *IBM Tivoli Workload Scheduler: Planning and Installation Guide*. The Tivoli Workload Scheduler components and their information flows are indicated in yellow, the IBM Tivoli Monitoring components and their information flows are indicated in blue.

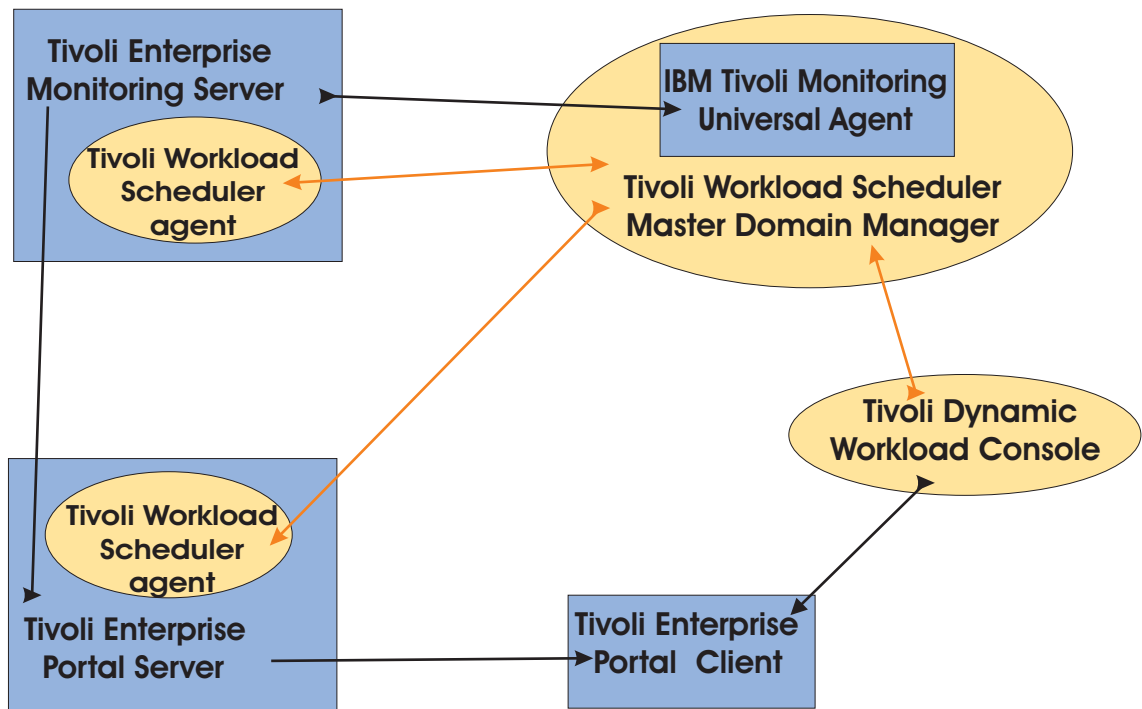


Figure 6. Information flow

These are the descriptions of the main information flows:

Information about the Tivoli Workload Scheduler master domain manager state and health

A script data provider runs on the Tivoli Workload Scheduler master domain manager to collect information about errors, anomalies, and threshold conditions. This information is saved in a file that is parsed by the Tivoli Universal Agent. The result of the parsing is retrieved on timely

basis by the Tivoli Enterprise Monitoring Server and then displayed on the Tivoli Enterprise Portal client interface.

Information about the processing of the scheduled backup activity

The Tivoli Workload Scheduler engines installed on the Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server run jobs to perform the requested the backup activity on these two servers. For specific states (for example HOLD) of a job and job stream that might represent an issue for a successful backup activity completion, events are triggered from the Tivoli Workload Scheduler agent where the job or job stream is processing.

Forwarding of selected events about the job processing status and issues to the Tivoli Enterprise Portal Server

Each time one of these events is triggered, it is saved by the Tivoli Workload Scheduler master domain manager, parsed by the Tivoli Universal Agent, retrieved by the Tivoli Enterprise Monitoring Server, and then passed to the Tivoli Enterprise Portal Server.

Situation events informing about the specific job or job stream issue are displayed on the Tivoli Enterprise Portal client user interface.

The user logged into the Tivoli Enterprise Portal Client user interface sees the situation event on the navigation tree informing about the specific job or job stream issue. The severity of the situation event is specified when setting up the Tivoli Enterprise Monitoring Server.

The Dynamic Workload Console is launched from the Tivoli Enterprise Portal client user interface to view and solve the issue

The user logged into the Tivoli Enterprise Portal client user interface selects the job or job stream instance marked with the situation event and launches the Dynamic Workload Console to access that specific job or job stream details and fix the problem, to ensure the backup activity is successfully completed within the selected time frame. When the problem has been solved, an event informing about the new state of the job or job stream is generated on the Tivoli Workload Scheduler workstation processing the job and the situation event is removed from the Tivoli Enterprise Portal navigation tree.

Setting up the environment

For details about how to run the steps listed in this section to configure the integration with Tivoli Enterprise Portal, refer to *Integrating with other IBM Tivoli Products - Integrating with Tivoli Monitoring*.

When the required software is installed, you must complete the following tasks before starting the scenario:

To enable IBM Tivoli Monitoring to monitor Tivoli Workload Scheduler master domain manager processes and data to look for errors, anomalies, and threshold conditions.

On the Tivoli Workload Scheduler master domain manager:

1. Install the Tivoli Universal Agent.
2. Run the ITMconfig script to configure the Universal Agent to monitor the Tivoli Workload Scheduler master domain manager.
3. Run the ITMCreateSituations script to configure the Tivoli Enterprise Portal Console to create and display the Tivoli Workload Scheduler situations.

To enable Tivoli Workload Scheduler events for all the monitored objects (jobs and job streams) to be passed to the Tivoli Enterprise Portal client user interface.

On the Tivoli Workload Scheduler master domain manager:

1. Run the TEPConfig script to enable monitoring by the Tivoli Enterprise Portal.
2. Tivoli Workload Scheduler creates and writes events for all the monitored objects (jobs and job streams) into a predefined event file that is accessed by the Tivoli Universal Agent at a set interval. Make sure that the `/TWS_home/BmEvents.conf` file is set to create and write the information about the following events:

Job is in abend state (event 101)
 Job is in hold (event 109)
 Job stream hold (event 159)

3. Recycle the Tivoli Workload Scheduler engine.

On the Tivoli Enterprise Monitoring Server:

Create situations by assigning the highest severity (Critical) when the following events are detected:

Job is in abend state (event 101)
 Job is in hold (event 109)
 Job stream hold (event 159)

When you complete these steps, the Tivoli Universal Agent is ready to retrieve and parse the logged events from the event file. The monitored data is then retrieved from the Tivoli Universal Agent by the Tivoli Enterprise Monitoring Server and displayed to the Tivoli Enterprise Portal.

To run the critical activity, the daily backup of the IBM Tivoli Monitoring data and configuration, using Tivoli Workload Scheduler

On the Tivoli Enterprise Monitoring Server and on the Tivoli Enterprise Portal Server:

Install a Tivoli Workload Scheduler engine

On the Tivoli Workload Scheduler master domain manager:

1. Create a job stream scheduled to run daily on the Tivoli Workload Scheduler engine installed on the Tivoli Enterprise Monitoring Server system during the time frame dedicated to the backup operations. The job stream must run in sequence the steps needed to run the backup of the Tivoli monitoring configuration and data. In this scenario you use a job stream named **BACKUP** containing such a sequence of jobs:

Table 14. Jobs implementing the backup procedure

Job name	Operation to run	Dependencies	Tivoli Workload Scheduler engine where the job runs
JOB1	Stop Tivoli Enterprise Portal Server	None	Engine installed on Tivoli Enterprise Portal Server system
JOB2	Stop Tivoli Enterprise Monitoring Server	JOB1 completed	Engine installed on Tivoli Enterprise Monitoring Server system

Table 14. Jobs implementing the backup procedure (continued)

Job name	Operation to run	Dependencies	Tivoli Workload Scheduler engine where the job runs
JOB3	tar -cvf <i>TEMS_installdir</i>	JOB2 completed. Needed disk space available on Tivoli Enterprise Monitoring Server system.	Engine installed on Tivoli Enterprise Monitoring Server system
JOB4	tar -cvf <i>TEPS_installdir</i>	JOB1 completed. Needed disk space available on Tivoli Enterprise Portal Server system.	Engine installed on Tivoli Enterprise Portal Server system
JOB5	db2backup <i>TEPS_db</i>	JOB1 completed. Needed disk space available on Tivoli Enterprise Portal Server system.	Engine installed on Tivoli Enterprise Portal Server system
JOB6	Start Tivoli Enterprise Monitoring Server	JOB3 completed	Engine installed on Tivoli Enterprise Monitoring Server system
JOB7	Start Tivoli Enterprise Portal Server.	JOB6, JOB4 and JOB5 completed	Engine installed on Tivoli Enterprise Portal Server system

The job stream and its jobs must be marked as *Monitored*. By doing so you ensure that, whenever an event among those specified in the `BmEvents.conf` file occurs on one of these jobs and job stream, it is parsed by the Tivoli Universal Agent, retrieved by the Tivoli Enterprise Monitoring Server and displayed on the Tivoli Enterprise Portal interface.

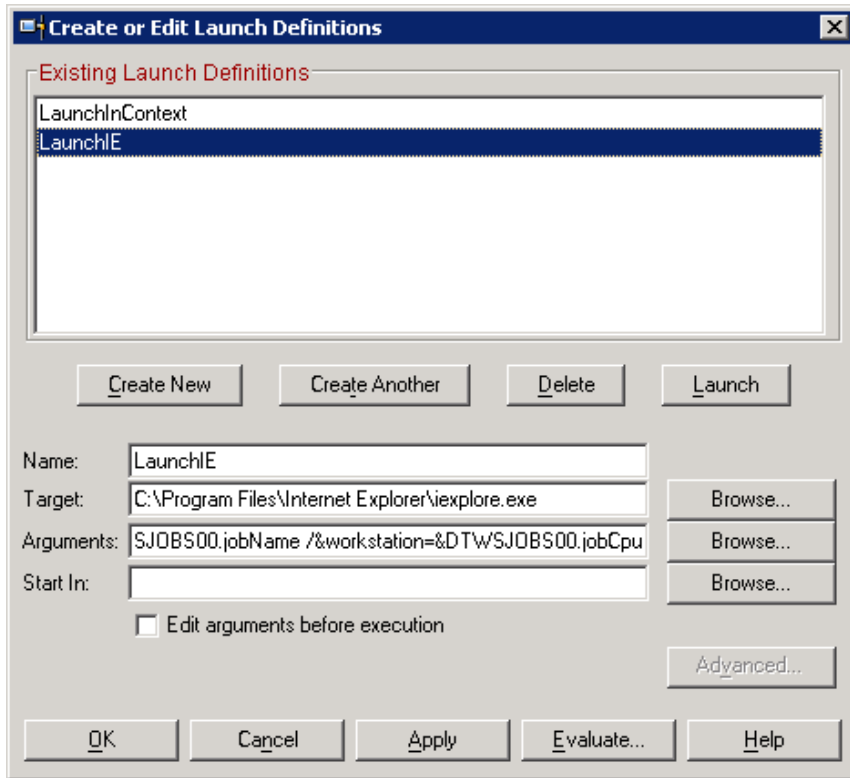
For details about how to run the steps to configure the jobs and job stream definitions refer to *IBM Tivoli Workload Scheduler: User's Guide and Reference*.

2. Run **JnextPlan** to add the new job stream and its jobs to the current plan.

To access specific job or job stream details, enable the Dynamic Workload Console launch in context from the Tivoli Enterprise Portal interface

In the navigation tree of the Tivoli Enterprise Portal client:

1. From the **Universal Agent** drill down to both the **JOBS** and the **JOBSTREAMS** entries.
2. Click a job or job stream instance and select **Create or Edit Launch Definitions**.
3. Select **Create New**.



4. Specify for each Launch Definition the following information:
- The name
 - The full path of the browser executable file to use when running the Dynamic Workload Console.
 - The URL to open using this format:

```
https://tdwc_hostname:tdwc_port/ibm/action/
launch?pageID=com.ibm.tws.WebUI.External.navigation/
&disableNavArea=false/&showNavArea=false/
&action=action/&hostname=engine_hostname/
&port=engine_port/&jobstream=TEP_argument_JOBSTREAM,/
&job=TEP_argument_JOB,/ &workstation=TEP_argument_CPU/
&schedtime=TEP_argument_SCHEDTIME
```

In this URL specify the following required settings:

tdwc_hostname

The hostname of the machine where Dynamic Workload Console is installed.

tdwc_port

The port number used to connect to Dynamic Workload Console (the default value is 29043 for https, 29060 for http).

action The value indicating the task that runs on Dynamic Workload Console on the selected object:

BrowseJobs

To run a query on jobs.

BrowseJobStreams

To run a query on job streams.

engine_hostname

The name of the Tivoli Workload Scheduler engine.

engine_port

The port number of the Tivoli Workload Scheduler engine.

The remaining settings:

- *TEP_argument_JOBSTREAM*
- *TEP_argument_JOB*
- *TEP_argument_CPU*
- *TEP_argument_SCHEDTIME*

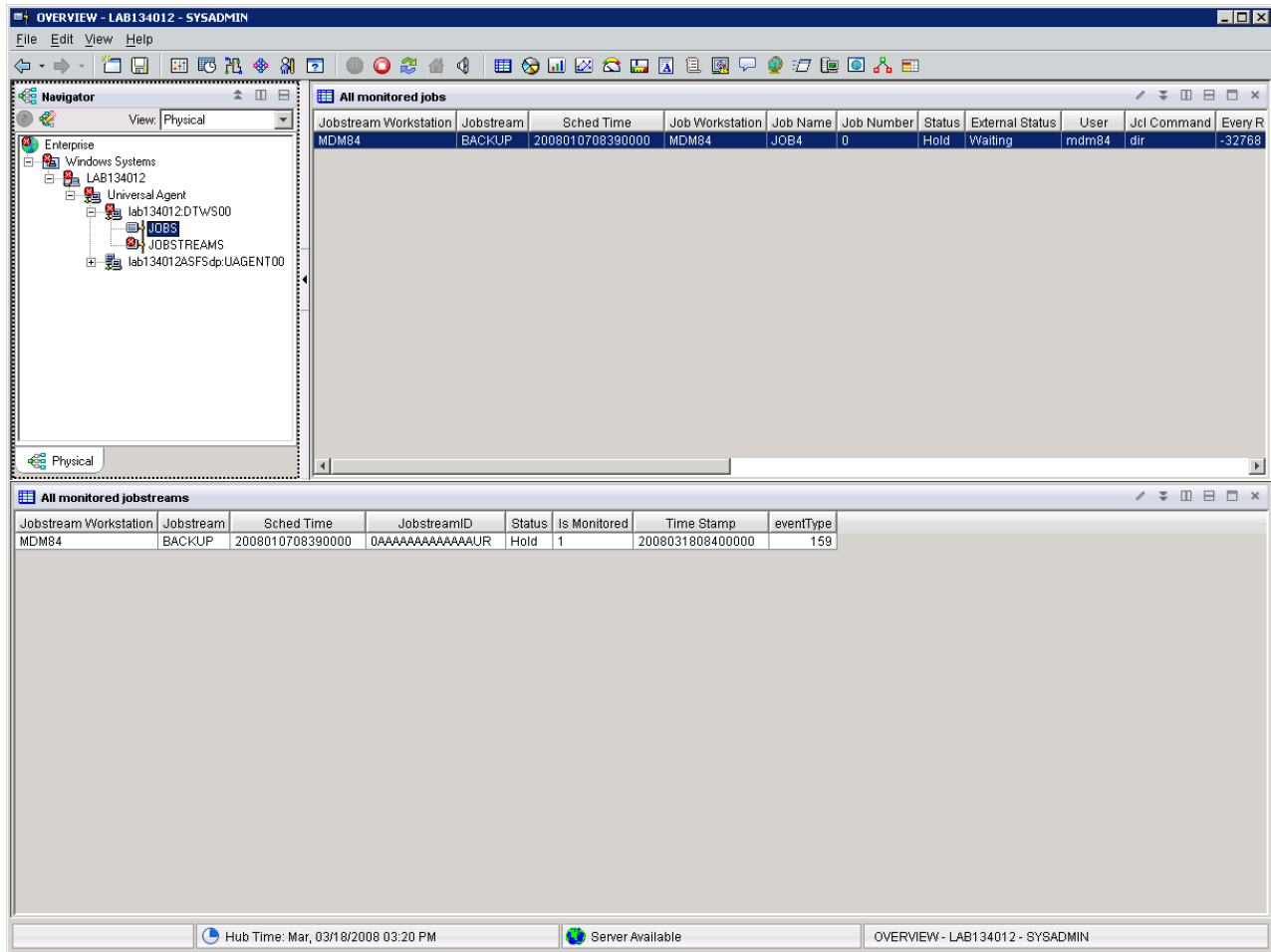
must be associated to Tivoli Enterprise Portal variables to be automatically filled in at run time with the specific values of the job or job stream selected on the Tivoli Enterprise Portal client interface. To do this, from the **Create or Edit Launch Definitions** panel, click the **Browse** button that appears to the right of the **Arguments** field. Select **Substitutable Items** and then select the Tivoli Workload Schedule variables.

5. To start Dynamic Workload Console, click **Evaluate**.

Running the scenario

This section lists the steps to run to complete the scenario.

In this scenario, the daily backup stops running because there is a lack of required free disk space available on the Tivoli Enterprise Portal Server system. The **BACKUP** job stream stops processing and remains in HOLD state. As a result, in the Tivoli Enterprise Portal client interface you see a red icon displayed in the navigation tree as follows:



Perform these steps to run the scenario:

1. Drill down the tree to open the **JOBSTREAM** view marked with the red icon.
2. Click the job stream **BACKUP** and select **Launch**.
3. In the Existing Launch Definition pane, select the Tivoli Workload Dynamic Console launch definition that you created to run the **BrowseJobStreams** action.
4. Enter the Tivoli Workload Dynamic Console credentials. A panel showing the details about that job stream run is displayed.
5. Access the jobs in that job stream by clicking **Jobs**. In the list, you can see that **JOB4** is in **HOLD**. Select that job and click **Dependencies** to see which unresolved dependency is holding up the job. You see that a resource dependency caused by a shortage of available disk space needed to save the tar copy on the Tivoli Enterprise Portal Server system prevents **JOB4** from running.
6. Either free the required disk space or ask the Tivoli Monitoring administrator to do so.
7. Run a refresh of the Dynamic Workload Console panel and check that the new job state is **READY**. Also check that the **BACKUP** job stream state is **READY**.
8. Return to the Tivoli Enterprise Portal client interface. As soon as the events information is parsed by the Tivoli Universal Agent and retrieved by the Tivoli Enterprise Monitoring server, you can see that the critical state icon disappears from the navigation tree.

9. All remaining jobs included in the **BACKUP** job stream complete running and the job stream ends successfully.

Chapter 2. Integrating with Tivoli NetView

How Tivoli Workload Scheduler on UNIX integrates with Tivoli NetView for AIX.

This chapter describes how Tivoli Workload Scheduler on UNIX integrates with Tivoli NetView for AIX version 7.1x. It is divided into the following sections:

- “Tivoli Workload Scheduler/NetView overview”
- “Installing the integration software”
- “Setting up Tivoli Workload Scheduler/NetView” on page 39
- “Menu actions” on page 40
- “Events” on page 41
- “Tivoli Workload Scheduler/NetView configuration files” on page 44
- “Tivoli Workload Scheduler/NetView configuration options” on page 47
- “Unison software MIB” on page 48
- “Tivoli Workload Scheduler/NetView program reference” on page 50

For a description of Tivoli NetView, refer to the Tivoli NetView documentation.

Tivoli Workload Scheduler/NetView overview

Tivoli Workload Scheduler/NetView consists of manager and agent software. The manager runs on NetView management nodes and the agent runs on managed nodes. All the nodes must have Tivoli Workload Scheduler installed. The manager polls its agents periodically to obtain information about scheduler processing. If the information returned during a poll is different from that of the preceding poll, the color of the corresponding symbol changes to indicate a state change. After you take action to remedy condition, the state of the corresponding symbol returns to normal at the next poll. The process that runs on the management node is called mdemon.

Agents generate SNMP traps to inform the manager of asynchronous events, such as job abends, stuck schedules, and restarted scheduler processes. Although polling and traps are functionally independent, the information that accompanies a trap can be correlated with symbol state changes. If, for example, a scheduled job abends, the symbol for the workstation changes color and a job abend trap is logged in the NetView event log. By scanning the log, you can isolate the problem and take the appropriate action.

The muser process runs commands issued by a NetView user, and updates the user's map. A muser is started for each NetView user whose map has the Tivoli Workload Scheduler/NetView application activate.

Installing the integration software

Tivoli Workload Scheduler/NetView is delivered as part of Tivoli Workload Scheduler. Before installing Tivoli Workload Scheduler/NetView, make sure that Tivoli Workload Scheduler is correctly installed on the management node (server and clients) and on each managed node and make sure it is successfully scheduling and tracking jobs.

Use the customize script to install the integration software. The customize script has the following syntax:

```
customize [-uname name] [-prev3] [-noinst] [-client] [-manager host ]
```

where:

[-uname name]

Tivoli Workload Scheduler user name.

-prev3 Include this option if your version of NetView is prior to version 3.

-noinst

Do not overwrite existing NetView configuration files.

-client For NetView version 6.x and later, include this option for management clients.

-manager

The host name of the management node. For NetView version 6.x and above, this is the host name of the NetView server. This is required for managed nodes and NetView clients. Do not use this option on the management node or NetView server.

Installing on managed nodes and NetView clients

The management node can also be a managed node. For the management node or NetView server, skip this step and perform step “Installing on the management node or NetView server.”

1. Make certain that no Tivoli Workload Scheduler processes are running. If necessary, issue a conman shutdown command.
2. Log in as root.
3. For managed nodes, including those that are also NetView clients that are not used to manage Tivoli Workload Scheduler, run the customize script as follows:

```
/bin/sh <TWShome>/OV/customize -manager host
```

where:

host is the host name of the management node.

4. For NetView clients that are used to manage Tivoli Workload Scheduler, run customize as follows:

```
/bin/sh <TWShome>/OV/customize -client [-manager host]
```

where:

host is the host name of the management node.

5. Run StartUp:

```
<TWShome>/StartUp
```

Installing on the management node or NetView server

1. Make certain that no Tivoli Workload Scheduler processes are running. If necessary, issue a conman shutdown command.
2. Log in as root.
3. Run the customize script as follows:

```
/bin/sh <TWShome>/OV/customize
```
4. If you do not want the Tivoli Workload Scheduler/NetView agent to run on this node, edit <TWShome>/StartUp, and remove the run of agent.

5. If you want Tivoli Workload Scheduler to run on this node, run StartUp:
`<TWShome>/StartUp`
 6. Start the Tivoli Workload Scheduler/NetView daemon (mdemon) as follows:
`/usr/0V/bin/ovstart Unison_Maestro_Manager`
- or, for NetView versions below 3, stop and start as follows:
- `/usr/0V/bin/ovstop`
-
- `/usr/0V/bin/ovstart`

Setting up Tivoli Workload Scheduler/NetView

To setup Tivoli Workload Scheduler/NetView, perform the following steps:

1. On each managed node, enter the host name of the management node in the user's \$HOME/.rhosts file.
2. Add a user definition to the scheduler security file. For more information about Tivoli Workload Scheduler security, refer to the *IBM Tivoli Workload Scheduler Reference Guide*.
3. On the management node, run NetView.
4. Select **Describe Map** from the File menu.
5. Select **Maestro-Unison Software (c)** from the Configurable Applications list, and click **Configure For This Map**.
6. Click **True** under Enable Maestro for this map.
7. Click **Verify**.
8. Click **OK** to close the Configuration dialog box. Click **OK** to close the Map Description dialog box.

If you want to use the MIB browser, perform the following steps:

1. Select **Load/Unload MIBs:SNMP** from the Options menu.
2. Click **Load**.
3. When Load MIB From File opens, type:
`/usr/0V/snmp_mibs/Maestro.mib`
 in MIB File to Load. Click **OK**.
4. Click **Close**.

If the management node is not also a Tivoli Workload Scheduler managed node, or if you manage more than one Tivoli Workload Scheduler network, use the NetView object description function to identify the managed nodes where Tivoli Workload Scheduler/NetView agents are running. To identify the managed nodes, perform the following steps:

1. Move down the IP Internet tree to the IP segment submap showing all the nodes.
2. Select a node where a Tivoli Workload Scheduler/NetView agent is running. Press **Ctrl-O** to open the Object Description dialog.
3. On the Object Description dialog, select **General Attributes** from the Object Attributes list, and click **View/Modify Object Attributes**.
4. On the Attributes for Object dialog, click **True** under the isUTMaestroAgent attribute.

5. Click **OK** to close the Attributes for Object dialog. Click **OK** to close the Object Description dialog.
6. Repeat steps for each node where a Tivoli Workload Scheduler/NetView agent is running.
7. Return to the Root submap.
8. Select **Tivoli Workload Scheduler**, from the Tools menu.
9. Select **Re-discover**.
10. When the Unison Software(c) symbol appears, double-click it to open the Unison Software(c) submap displaying a symbol for each Tivoli Workload Scheduler network. Double-click a network symbol to open a Network submap.

On the master domain issue a `conman start@` command to start Tivoli Workload Scheduler in the network. This can be done in NetView on the Tivoli Workload Scheduler Network submap as follows:

1. Select all of the nodes in the network.
2. From the Tools menu, select **Tivoli Workload Scheduler**.
3. Select **Start**.

Menu actions

To use Tivoli Workload Scheduler/NetView menu actions, select **Tivoli Workload Scheduler** from the **Tools** menu. These actions are also available from the object context menu by right clicking a symbol.

The menu actions are:

View Open a child submap for a Tivoli Workload Scheduler/NetView symbol. Choosing View after selecting a workstation symbol on the submap opens the monitored processes submap. Choosing View after selecting a workstation symbol on the IP node submap returns to the Tivoli Workload Scheduler network submap.

Master conman

Run the conman command-line on the Tivoli Workload Scheduler master. Running on the master lets you run conman commands (except shutdown) for any workstation in the network. For information about conman commands, see *IBM Tivoli Workload Scheduler: User's Guide and Reference*.

Acknowledge

Acknowledge the status of selected symbols. When acknowledged, the status of a symbol returns to normal. It is not necessary to acknowledge critical or marginal states for a monitored process symbol, as it returns to normal when the monitored process itself is running. Acknowledge critical or marginal states for workstation symbols, otherwise they do not return to normal.

Conman

Run the conman command-line on the selected Tivoli Workload Scheduler workstations. Running on a workstation other than the master, lets you run all conman commands on that workstation only. For information about conman commands, see *IBM Tivoli Workload Scheduler: User's Guide and Reference*. For an extended agent, conman is run on its host.

Start Issue a conman start command for the selected workstations. By default, the command for this action is:

```
remsh %H %P/bin/conman 'start %c'
```

Down (stop)

Issue a conman stop command for the selected workstations. By default, the command for this action is:

```
remsh %H %P/bin/conman 'stop %c'
```

StartUp

Run the Tivoli Workload Scheduler StartUp script on the selected workstations. By default, the command for this action is:

```
remsh %h %P/StartUp
```

For an extended agent, conman is run on its host.

Rediscover

Locate new agents and new Tivoli Workload Scheduler objects, and update all Tivoli Workload Scheduler/NetView sub-maps.

Note: Run Rediscover each time you change the Tivoli Workload Scheduler workstation configuration.

The substituted parameters in the command-lines are:

- %c** The Tivoli Workload Scheduler workstation name of a selected workstation symbol.
- %D** The current DISPLAY name.
- %h** The host name of a selected workstation symbol.
- %H** The host name of the Tivoli Workload Scheduler master.
- %p** The process name of a selected process symbol, or “MAESTRO” if it is not a process.
- %P** The maestro user's home directory (usually /usr/lib/maestro).

Events

Events 1-53 indicate the status of critical processes that are monitored by the Tivoli Workload Scheduler/NetView agents, including the agents themselves (event 1). Events 101-25) indicate the status of the job scheduling activity.

The listed events can result in SNMP traps generated by the Tivoli Workload Scheduler/NetView agents. Whether or not traps are generated is controlled by options set in the configuration files of the agents. See “Tivoli Workload Scheduler/NetView configuration files” on page 44 for more information.

The Additional Actions column lists the actions available. The actions can be initiated by selecting **Additional Actions** from the **Options** menu, then selecting an action from the Additional Actions panel.

Note: You need the appropriate Tivoli Workload Scheduler security access to perform the chosen action. See *IBM Tivoli Workload Scheduler: Administration Guide*.

Table 15. Tivoli Workload Scheduler/NetView events

Trap #	Name	Description	Additional Actions
1 *	uTtrapReset	The magent process was restarted.	
51	uTtrapProcessReset	A monitored process was restarted. This event is reported by default in the BmEvents.conf file	
52 *	uTtrapProcessGone	A monitored process is no longer present.	
53 *	uTtrapProcessAbend	A monitored process abended.	
54 *	uTtrapXagentConnLost	The connection between a host and xagent has been lost.	
101 *	uTtrapJobAbend	A scheduled job abended.	Show Job Rerun Job Cancel Job
102 *	uTtrapJobFailed	An external job is in the <i>error</i> state.	Show Job Rerun Job Cancel Job
103	uTtrapJobLaunch	A scheduled job was launched successfully.	Show Job Rerun Job Cancel Job
104	uTtrapJobDone	A scheduled job finished in a state other than ABEND.	Show Job Rerun Job Cancel Job
105*	uTtrapJobUntil	A scheduled job's UNTIL time has passed, it will not be launched.	Show Job Rerun Job Cancel Job
111	TrapJobCant	A scheduled job could not be launched.	Show Job Rerun Job Cancel Job
151 *	uTtrapSchedAbend	A schedule ABENDED.	Show Schedule Cancel Schedule
152 *	uTtrapSchedStuck	A schedule is in the STUCK state.	Show Schedule Cancel Schedule
153	uTtrapSchedStart	A schedule has started execution.	Show Schedule Cancel Schedule
154	uTtrapSchedDone	A schedule has finished in a state other than ABEND.	Show Schedule Cancel Schedule
155*	uTtrapSchedUntil	A schedule's UNTIL time has passed, it will not be launched.	Show Schedule Cancel Schedule
201 *	uTtrapGlobalPrompt	A global prompt has been issued.	Reply
202 *	uTtrapSchedPrompt	A schedule prompt has been issued.	Reply
203 *	uTtrapJobPrompt	A job prompt has been issued.	Reply
204 *	uTtrapJobRerunPrompt	A job rerun prompt has been issued.	Reply
251	uTtrapLinkDropped	The link to a workstation has closed.	Link
252 *	uTtrapLinkBroken	The link to a workstation has closed due to an error.	Link

Table 15. Tivoli Workload Scheduler/NetView events (continued)

Trap #	Name	Description	Additional Actions
261	TWS_Stop_Monitoring	Sent when the monitoring status of an agent is set to off (for stopmon command or because the agent is unable to send events to the event processing server)	
262	TWS_Start_Monitoring	Sent when the monitoring status of an agent is set to on (for startmon command or because the agent has re-started to send events to the event processing server)	
* These traps are enabled by default.			

Polling and SNMP traps

Because SNMP uses an unreliable transport protocol (UDP), Tivoli Workload Scheduler/NetView does not rely on SNMP traps to indicate the status of its symbols. The manager polls its agents periodically, requesting specific MIB values. The returned values are compared with those returned by the previous poll, and differences are indicated as status changes in Tivoli Workload Scheduler/NetView symbols. The default polling interval is one minute. See “Tivoli Workload Scheduler/NetView configuration options” on page 47 for information about changing the polling interval.

To obtain critical process status, the manager polls all of its agents. For job scheduling status, the manager determines which of its agents is most likely to have the required information, and polls only that agent. The choice is made in the following order of precedence:

1. The agent running on the Tivoli Workload Scheduler master
2. The agent running on a Tivoli Workload Scheduler backup master
3. The agent running on any Tivoli Workload Scheduler fault-tolerant agent that has full status on in its workstation definition

Enabling traps provides the following advantages:

1. Event-specific variables are included with each trap
2. Traps are logged in NetView's event log.

If job abend traps (101) are enabled, for example, sufficient information is collected to identify an abended job, its schedule, and the workstation on which it runs. This is useful when deciding what actions to take to remedy a problem.

You might choose to disable some or all of the Tivoli Workload Scheduler/NetView traps for the following reasons:

1. To reduce network traffic
2. To prevent NetView user's from receiving logged events that are not relevant

For more information about the Unison Software's enterprise-specific traps and their variables, see “Re-configuring enterprise-specific traps” on page 48.

Tivoli Workload Scheduler/NetView configuration files

On each managed node (each node running a Tivoli Workload Scheduler/NetView agent), the selection of events and how they are reported is controlled by setting variables in two configuration files:

- The BmEvents configuration file controls the reporting of job scheduling events (101-252 in Table 15 on page 42) by the mailman and batchman production processes. These events are passed on to the agent, which might convert them to SNMP traps, depending on the settings in its configuration file.
- The MAgent configuration file controls reporting by the Tivoli Workload Scheduler/NetView agent, magent. Events selected in this file are turned into SNMP traps, which are passed to NetView by the Tivoli Workload Scheduler/NetView manager, mdemon, on the management node. The traps can also be processed by other network management systems.

The BmEvents configuration file

The BmEvents configuration file is named <TWShome>/BmEvents.conf. Use it to configure Tivoli Workload Scheduler production processes on each workstation that has an agent installed. Its contents are described below.

comment

A comment line.

OPTIONS=MASTER|OFF

If the value is set to MASTER then all job scheduling events gathered by that workstation are reported. If that workstation is the master domain manager or the backup master domain manager with full status on, then all scheduling events from the scheduling environment are reported. If the value is set to OFF, no job scheduling events are reported from that workstation. If commented, it defaults to MASTER on the master domain manager workstation, while it allows to report all job scheduling events regarding that workstation only on a workstation different from the master domain manager.

EVENT= *n* [*n* ...]

The list of events to be reported. Event numbers must be separated by at least one space. If omitted, the events reported by default are:

51 101 102 105 151 152 155 201 202 203 204 251 252

Event 51 causes mailman and batchman to report the fact that they were restarted. Events 1, 52, and 53 are not valid in this file (see “The MAgent configuration file” on page 45).

If the EVENT parameter is included, it completely overrides the defaults. To remove only event 102 from the list, for example, you must enter the following:

EVENT=51 101 105 151 152 155 201 202 203 204 251 252

See Table 15 on page 42 for a description of events.

PIPE=*filename*

If set, job scheduling events are written to a FIFO file. To have events sent to the Tivoli Workload Scheduler/NetView agent, the setting must be:

PIPE=MAGENT.P

CHSCHEd=HIGH | LOW

When set to HIGH, batchman sends an event for any schedule status transaction. When set to LOW, batchman only tracks the initial schedule status transactions. For the lifetime of schedule jobs no change of status is reported until the final state is reached. When a job has more than one final state, an event is sent for each. For example, a schedule completes with an ABEND state and event 151 is sent (schedule abended). The job is then rerun and completes successfully. The schedule is completed with a SUCC state and event 154 is sent (schedule completed). The default is HIGH. Table 16 lists the events that are filtered by CHSCHEd when it is set to LOW.

Table 16. Events filtered by CHSCHEd

Event	Description	Filtered on LOW
151	Schedule abended	NO
152	Schedule is stuck	NO
153	Schedule started	YES
154	Schedule ended	NO
155	Until time expired onuntil = suppr	NO
156	Schedule submitted	YES
157	Schedule cancelled	NO
158	Schedule ready	YES
159	Schedule hold	YES
160	Schedule extrn	YES
161	Schedule is cancel pending	NO
162	Schedule properties changed	YES
163	Schedule is late	NO
164	Until time expired onuntil = continue	NO
165	Until time expired onuntil = cancel	NO

A BmEvents configuration file is included with the Tivoli Workload Scheduler software. It contains several comment lines, and a single parameter setting:

```
PIPE=MAGENT.P
```

This causes events to be reported as follows:

- If installed on the master, it will report all job scheduling events (101-252) for all workstations in the network. If installed on any other workstation, no job scheduling events will be reported. The process restart event (51) is reported regardless of the workstation type.
- The following events are reported:
51 101 102 105 151 152 155 201 202 203 204 251 252
- Event information is written to a FIFO file named MAGENT.P, which is read by the Tivoli Workload Scheduler/NetView agent.

The MAgent configuration file

The MAgent configuration file is named <TWSHome>/MAgent.conf. Use it to configure the agent on each workstation. Its contents are described below.

comment

A comment line.

OPTIONS=MASTER|OFF

If set to MASTER, the agent on this workstation sends the job scheduling events read from the MAGENT.P file as SNMP traps. If set to OFF, no job scheduling traps are generated by this workstation. If omitted, it defaults to MASTER on the master, and OFF on other workstations.

This variable is required only if the master will not be used to generate job scheduling traps for the network. For example, if the master is not a managed node (no agent is installed), you should set this variable to MASTER on a backup master that has an agent installed.

EVENT= *n* [*n* ...]

The list of events to be sent as SNMP traps. With the exception of events 1, 52, and 53, traps will not be generated unless the corresponding events are turned on in the BmEvents configuration file. Event numbers must be separated by at least one space. If omitted, the events sent as traps by default are:

1 52 53 54 101 102 105 151 152 155 201 202 203 204 252

Event 1 (magent restarted) cannot be turned off.

If this parameter is included, it completely overrides the defaults. To remove only event 102 from the list, for example, you must enter the following:

EVENT=1 52 53 54 101 105 151 152 155 201 202 203 204 252

See Table 15 on page 42 for a description of events.

+name [*pidfilename*]

By default, the list of processes monitored by the Tivoli Workload Scheduler/NetView agent contains the following processes: magent, netman, mailman, batchman, jobman, all mailman servers, all writers, and all extended agent connections. Use this syntax to add processes to the list. If it is not a Tivoli Workload Scheduler process, you must include its PID file name. Some examples are:

+SENDMAIL /etc/sendmail.pid

+SYSLOG /etc/syslogd.pid

-name Use this syntax to remove processes from the list of monitored processes. To remove writer processes, use this form:

- cpuid :writer

For example, to remove the writers for all workstations with ids starting with SYS, enter:

-SYS@:WRITER

To remove all writers, enter:

-@:WRITER

To remove mailman servers 5 and A, enter:

-SERVER5

-SERVERA

To remove all mailman servers, enter:

-SERVER@

An MAgent configuration file is included with the Tivoli Workload Scheduler/NetView software. It contains only comment lines with no parameters set. This causes SNMP traps to be generated as follows:

- If installed on the master, traps are generated for job scheduling events (101-252) on all workstations in the network. If installed on any other workstation, no job scheduling traps are generated.
- The following events result in SNMP traps:
1 52 53 54 101 102 105 151 152 155 201 202 203 204 252
- The following processes are monitored: magent, netman, mailman, batchman, jobman, all mailman servers, all writers, and all extended agent connections.

Monitoring writers and servers

writer and mailman server processes are started and stopped when workstations are linked and unlinked. Their transitory nature and the resulting number of status changes in NetView can cause confusion, particularly in large networks where linking and unlinking is common. For this reason, you can remove writer and mailman server processes from the list of monitored processes.

Tivoli Workload Scheduler/NetView configuration options

Tivoli Workload Scheduler/NetView submaps, symbols, and objects can be modified like others in NetView. The following topics describe some specific configuration options for Tivoli Workload Scheduler/NetView.

Agent scan rate

By default, the agents scan and update the status of their monitored processes every 60 seconds. To change the rate:

1. Login on the managed node and edit the file <TWShome>/StartUp.
2. Add the -timeout option to the magent command line.

For example, to change the rate to 120 seconds, make the following change:

```
<TWShome>/bin/magent -peers hosts -timeout 120
```

Manager polling rate

The mdemon manager polls its agents to retrieve status information about the managed nodes. The rate is defined in the file /usr/OV/lrf/Mae.mgmt.lrf on the management node. Unless otherwise specified, the polling rate defaults to 60 seconds.

To change the rate:

1. Edit the file to add the -timeout option to the mdemon command line. For example, to change the rate to 120 seconds, make the following change:
Unison_Software_Maestro_Manager: <TWShome>/bin/mdemon:
OVs_YES_START:pmd,ovwdb:-pmd,-timeout,120:OVs_WELL_BEHAVED
2. After making a change, delete the old registration by running the ovdlobj command.
3. Register the manager by running the ovaddobj command and supplying the name of the lrf file.

For more information, review the man pages for `ovaddobj(8)` and `lrf(4)`. See also “Configuring agents in NetView.”

Configuring agents in NetView

To change the configuration of Tivoli Workload Scheduler/NetView agents in NetView, follow these steps:

1. Move down the IP Internet tree to the IP Segment submap showing all the nodes.
2. Select a node where a Tivoli Workload Scheduler/NetView agent is running. Enter **Ctrl-O** to open the Object Description panel.
3. In the Object Description panel, select **Maestro - Unison Software(c)** from the Object Attributes list.
4. Click **View/Modify Object Attributes**.
5. On the Attributes for Object panel:
 - a. To ignore this agent altogether, click **False** under Does a Maestro agent exist on this cpu?.
 - b. To change the rate at which `mdemon` polls this agent, enter the number of seconds under Enter the number of seconds between polling. If this number is other than zero, it overrides the rate defined for the `mdemon` process (see “Manager polling rate” on page 47).
 - c. To close the Attributes for Object panel, click **Verify** and then **OK**.
6. To close the Object Description panel, click **OK**.

Configuring workstation status in NetView

To modify the way status is indicated for a Tivoli Workload Scheduler workstation symbol, follow these steps:

1. Select a workstation symbol on the Tivoli Workload Scheduler network submap.
2. To open the Object Description panel, enter **Ctrl-O**.
3. On the Object Description dialog, select **Tivoli Workload Scheduler** from the Object Attributes list.
4. Click **View/Modify Object Attributes**.
5. On the Attributes for Object dialog: Click **True** or **False** to ignore or recognize the various job scheduling events. For example, to ignore job abend events, click **True** under Tivoli Workload Scheduler should ignore JobAbend Events.
6. To close the Attributes for Object panel, click **Verify** and then **OK**.
7. To close the Object Description panel, click **OK**.

Unison software MIB

For a complete listing of the Unison Software enterprise MIB, see the file `TWShome/OV/Maestro.mib`.

Re-configuring enterprise-specific traps

The Tivoli Workload Scheduler/NetView enterprise-specific traps are configured with default messages that will serve most users' needs. To re-configure the traps,

choose Event Configuration from the Options menu. For instructions, refer to your NetView documentation or online help. It might also be helpful to review the man page for *trapd.conf*(4).

The enterprise-specific traps and their positional variables are listed in Table 17. Trap descriptions are listed in Table 15 on page 42.

Table 17 lists enterprise-specific traps.

Table 17. Enterprise-specific traps

Trap	Identifier	Positional variables
1 *	uTtrapReset	<ol style="list-style-type: none"> 1. Agent identifier number 2. Software version 3. Tivoli Workload Scheduler message string, if any
51 52 * 53 *	uTtrapProcessReset uTtrapProcessGone uTrapProcessAbend	<ol style="list-style-type: none"> 1. Process pid 2. Program name 3. Tivoli Workload Scheduler message string, if any
54 *	uTrapXagentConnLost	<ol style="list-style-type: none"> 1. Program name 2. Tivoli Workload Scheduler message string, if any
101 * 102 * 103 104 105 * 204 *	uTtrapJobAbend uTtrapJobFailed uTtrapJobLaunch uTtrapJobDone uTtrapJobUntil uTtrapJobRerunPrompt	<ol style="list-style-type: none"> 1. workstation name of the schedule. 2. Schedule name. 3. Job name. For jobs submitted with <i>at</i> or <i>batch</i>, if the name supplied by the user is not unique, this is the Tivoli Workload Scheduler-generated name, and the name supplied by the user appears as variable 7. 4. workstation name on which the job runs. 5. Job number (pid). 6. Job state, indicated by an integer: 1 (ready), 2 (hold), 3 (exec), 5 (abend), 6 (succ), 7 (cancl), 8 (done), 13 (fail), 16 (intro), 23 (abenp), 24 (succp), 25 (pend). 7. Job's submitted (real) name. For jobs submitted with <i>at</i> or <i>batch</i>, this is the name supplied by the user if not unique. The unique name generated by Maestro appears as variable 3. 8. User name under which the job runs. 9. Name of the job's script file, or the command it executes. White space is replaced by the octal equivalent; for example, a space appears as 040. 10. The rate at which an <i>every</i> job runs, expressed as <i>hhmm</i>. If <i>every</i> was not specified for the job, this is -32768. 11. Job recovery step, indicated by an integer: 1 (stop), 2 (stop after recovery job), 3 (rerun), 4 (rerun after recovery job), 5 (continue), 6 (continue after recovery job), 10 (this is the rerun of the job), 20 (this is the run of the recovery job). 12. An event timestamp, expressed as: <i>yyyymmddhhmmss 00</i> (that is, year, month, day, hour, minute, second, hundredths always zeroes). 13. The prompt number, or zero if there is no prompt. 14. The prompt text, or Tivoli Workload Scheduler error message.

Table 17. Enterprise-specific traps (continued)

Trap	Identifier	Positional variables
151 *	uTtrapSchedAbend	1. Workstation name of the schedule.
152 *	uTtrapSchedStuck	2. Schedule name.
153	uTtrapSchedStart	3. Schedule state, indicated by an integer: 1(ready), 2 (hold), 3 (exec), 4 (stuck), 5 (abend), 6 (succ), 7 (cancl).
154	uTtrapSchedDone	
155 *	uTtrapSchedUntil	4. Tivoli Workload Scheduler error message, if any.
201 *	uTtrapGlobalPrompt	1. Prompt name. 2. Prompt number. 3. Prompt text.
202 *	uTtrapSchedPrompt	1. Workstation name of the schedule. 2. Schedule name. 3. Prompt number. 4. Prompt text.
203 *	uTtrapJobPrompt	1. Workstation name of the schedule. 2. Schedule name. 3. Job name. 4. Workstation name of the job. 5. Prompt number. 6. Prompt text.
251 *	uTrapLinkDropped	1. The to workstation name. 2. Link state, indicated by an integer: 1 (unknown), 2 (down due to an unlink), 3 (down due to an error), 4 (up). 3. Tivoli Workload Scheduler error message. uTrapLinkDropped corresponds to link state 2) only.
252 *	uTrapLinkBroken	1. The to workstation name. 2. Link state, indicated by an integer: 1 (unknown), 2 (down due to an unlink), 3 (down due to an error), 4 (up). 3. Tivoli Workload Scheduler error message. uTrapLinkBroken corresponds to link state 3) only.
* These traps are enabled by default.		
Link states 1) unknown and 4) up are not used, as they are not relative to traps 251 and 252.		

Tivoli Workload Scheduler/NetView program reference

The following information is provided for those who want to run the Tivoli Workload Scheduler/NetView programs manually. The manager program, mdemon, is normally started with NetView as part of the ovstart sequence, and its run options are included in the /usr/OV/lrf/Mae.mgmt.lrf file. The agent program, magent, is normally started within the Tivoli Workload Scheduler StartUp script (<TWSHome>/bin/StartUp).

mdemon synopsis

```
mdemon [-timeout <secs>] [-pmd] [-port <port>] [-retry <secs>]
```

where,

-timeout

The rate at which agents are polled, expressed in seconds. The default is 60 seconds. See "Manager polling rate" on page 47 and "Configuring agents in NetView" on page 48 for more information about changing the rate.

-pmd

This option causes mdemon to run under NetView pmd (Port Map Demon). Otherwise, it must be run manually. This option is included by default in the file /usr/OV/lrf/Mae.mgmt.lrf file.

-port

For HP-UX agents only. This identifies the port address on the managed nodes on which the HP-UX agents will respond. The default is 31112.

-retry

The period of time mdemon waits before trying to reconnect to a non-responding agent. The default is 600 seconds.

magent synopsis

The syntax of magent is:

```
magent -peers <host> [, <host> [...]] [-timeout <secs> ] [-notraps] [-port <port>]
```

where:

-peers

For HP-UX agents only. This defines the hosts (names or IP addresses) to which the agent will send its traps. The default is 127.0.0.1 (loopback).

For AIX agents, the /etc/snmpd.conf file must be modified to define the hosts to which the agent will send its traps. To add another host, for example, duplicate the existing trap line and change the host name:

```
# This file contains Tivoli Workload Scheduler
# agent registration.
#
trap    public host1  1.3.6.1.4.1.736 fe
trap    public host2  1.3.6.1.4.1.736 fe
```

-timeout

The rate at which the agent checks its monitored processes, expressed in seconds. The default is 60 seconds.

-notraps

If included, the agent will not generate traps.

-port

For HP-UX agents only. This defines the port address on which this agent responds. The default is 31112.

Chapter 3. Integrating with Tivoli Enterprise Console

How Tivoli Workload Scheduler integrates with Tivoli Enterprise Console.

This chapter describes how Tivoli Workload Scheduler integrates with Tivoli Enterprise Console versions 3.8 and 3.9. It is divided into the following sections:

- “Configuring the Tivoli Enterprise Console adapter”
- “Configuring the Tivoli Enterprise Console server” on page 54
- “Event formats” on page 55
- “Re-loading monitoring data” on page 59
- “Job scheduling events” on page 61
- “Job scheduling events format” on page 69

Configuring the Tivoli Enterprise Console adapter

This section describes how to enable Tivoli Enterprise Console to receive Tivoli Workload Scheduler events.

The Tivoli Enterprise Console logfile adapter is used to relay events from the workstations in the scheduling environment to the Tivoli Enterprise Console event server. Depending on the workstation in your scheduling environment where you decide to install and configure the Tivoli Enterprise Console logfile adapter, you can have different events displayed in the event console.

When you have installed the Tivoli Enterprise Console logfile adapter on a workstation, a set of configuration steps must be performed to enable that adapter to manage the job scheduling events. For information about how to install the Tivoli Enterprise Console logfile adapter, refer to the *IBM Tivoli Enterprise Console Installation Guide*.

Use the **config_teclogadapter** script to configure the Tivoli Enterprise Console adapter installed on the Tivoli Workload Scheduler system that you want to monitor. Perform the following steps:

1. Set the environment variables for the Tivoli endpoint by running the `lcf_env` script.
2. Run the **config_teclogadapter** script to configure the adapter. For example:
`config_teclogadapter [-tme] PATH [Adapter ID] [TWS Installation Path]`

where:

-tme The Tivoli Enterprise Console adapter is a TME adapter.

PATH Specify the Tivoli Enterprise Console adapter directory when you did not specify the **-tme** option. Otherwise it is the endpoint directory.

Adapter ID

Specify the Tivoli Enterprise Console Adapter identifier (only for Tivoli Enterprise Console 3.9 and later). If you do not specify an ID, it is ignored.

TWS Installation Path

Specify the path where the Tivoli Workload Scheduler you want to monitor is installed.

The script performs the following configuration steps:

1. If no Tivoli Workload Scheduler installation path was specified, it uses the home directory where it is installed.
2. Copies the config/BmEvents.conf into the home directory if it does not already exist.
3. Configures the config/BmEvents.conf adding the list of events if not already specified and defines the event.log file as an event output.
4. Configures the configuration file of the Tivoli Enterprise Console adapter to read from the event.log file.
5. Appends the maestro.fmt file to the format file of the Tivoli Enterprise Console adapter and regenerate the cds file.
6. Restarts the Tivoli Enterprise Console adapter.

After you run the script, perform a **conman stop** and **conman start** to apply the changes.

Configuring the Tivoli Enterprise Console server

In addition to configuring the Tivoli Enterprise Console adapter, you must also configure the Tivoli Enterprise Console server.

Use the **config_tecserver** script to configure the Tivoli Enterprise Console server to enable the server to receive events from the Tivoli Enterprise Console adapter. It must be run on the system where the Tivoli Enterprise Console Server is installed or on a ManagedNode of the same TME network. On the Windows platform, a TME bash is required to run the script. For example:

```
config_tecserver.sh { -newrb <RuleBase name=""> <RuleBase
Path=""> -clonerb <RuleBase name=""> | -userb <RuleBase
name=""> }
<EventConsole> [TECUIServer host] USER PASSWORD
```

where:

-newrb

Specify a new RuleBase with the specified name and path.

-clonerb

Specify the rule base to be cloned into the new Rule base.

-userb Customize an already existing RuleBase.

EventConsole

Specify the EventConsole to be created and configured.

TECUIServer host

Specify the host name where the Tivoli Enterprise Console UI server is installed.

USER PASSWORD

Specify the user name and password needed to access the EventConsole.

The script performs the following configuration steps:

1. If specified, creates the new RuleBase from the cloned one.

2. Adds the Tivoli Workload Scheduler baroc events definition to the specified RuleBase.
3. Adds the Tivoli Workload Scheduler rules to the RuleBase.
4. Compile the RuleBase.
5. Put the RuleBase as the Active RuleBase.
6. Configures the specified EventConsole with Tivoli Workload Scheduler filters.
7. Restarts the Tivoli Enterprise Console server.

Event formats

Table 18 lists the engine event formats.

Table 18. Tivoli Workload Scheduler engine events format

Event	Number
mstReset	1
mstProcessGone	52
mstProcessAbend	53
mstXagentConnLost	54
mstJobAbend	101
mstJobFailed	102
mstJobLaunch	103
mstJobDone	104
mstJobUntil	105
mstJobSubmit	106
mstJobCancel	107
mstJobReady	108
mstJobHold	109
mstJobRestart	110
mstJobCant	111
mstJobSuccp	112
mstJobExtrn	113
mstJobIntro	114
mstJobWait	116
mstJobWaitd	117
mstJobSched	118
mstJobModify	119
mstJobLate	120
mstJobUntilCont	121
mstJobUntilCanc	122
mstSchedAbend	151
mstSchedStuck	152
mstSchedStart	153
mstSchedDone	154
mstSchedUntil	155

Table 18. Tivoli Workload Scheduler engine events format (continued)

Event	Number
mstSchedSubmit	156
mstSchedCancel	157
mstSchedReady	158
mstSchedHold	159
mstSchedExtrn	160
mstSchedCnpend	161
mstSchedModify	162
mstSchedLate	163
mstSchedUntilCont	164
mstSchedUntilCanc	165
mstGlobalPrompt	201
mstSchedPrompt	202
mstJobPrompt	203
mstJobRecovPrompt	204
mstLinkDropped	251
mstLinkBroken	252
mstDomainMgrSwitch	301

Positional event variables

This subsection defines the positional event variables.

Table 19. Positional variables for events 101-118,120-122,204 (job events)

Variable	Description
1	event number
2	schedule cpu
3	schedule id
4	job name
5	job cpu
6	job number
7	job status
8	real name (different from job name only for MPE jobs)
9	job user
10	jcl name (script name or command name)
11	every time
12	recovery status
13	time stamp (yyyymmddhhmm0000)
14	message number (not equal to zero only for job recovery prompts)
15	eventual text message (delimited by '\t')
16	record number

Table 19. Positional variables for events 101-118,120-122,204 (job events) (continued)

Variable	Description
17	key flag
18	effective start time
19	estimated start time
20	estimated duration
21	deadline time (epoch)
22	return code
23	original schedule name (schedule name for schedules not (yet) carried forward)
24	head job record number (different from record number for rerun/every jobs)
25	Schedule name
26	Schedule input arrival time (yyyymmddhhmm00)

Table 20. Positional variables for event 119 (job property modified)

Variable	Description
1	event number
2	schedule cpu
3	schedule id
4	job name
5	job cpu
6	job number
7	property type: StartTime = 2, StopTime = 3, Duration = 4, TerminatingPriority = 5, KeyStatus = 6
8	property value
9	record number
10	key flag
11	head job record number (different from record number for rerun/every jobs)
12	real name (different from job name only for MPE jobs)
13	original schedule name (schedule name for schedules not (yet) carried forward)
14	message number (not equal to zero only for job recovery prompts)
15	Schedule name
16	Schedule input arrival time (yyyymmddhhmm00)

Table 21. Positional variables for events 151-161, 163-165 (schedule events)

Variable	Description
1	event number
2	schedule cpu
3	schedule ID
4	schedule status
5	record number
6	key flag
7	original schedule name (schedule name for schedules not (yet) carried forward)
8	time stamp
9	Schedule name
10	Schedule input arrival time (yyyymmddhhmm00)

Table 22. Positional variables for event 162 (schedule property modified)

Variable	Description
1	event number
2	schedule cpu
3	schedule id
4	property type: StartTime = 2 StopTime = 3
5	property value
6	record number
7	original schedule name (schedule name for schedules not (yet) carried forward)
8	time stamp
9	Schedule name
10	Schedule input arrival time (yyyymmddhhmm00)

Table 23. Positional variables for event 202 (schedule prompt)

Variable	Description
1	event number
2	schedule cpu
3	schedule id
4	Schedule name
5	Schedule input arrival time (yyyymmddhhmm00)

Table 24. Positional variables for event 203 (job prompt)

Variable	Description
1	event number
2	schedule cpu

Table 24. Positional variables for event 203 (job prompt) (continued)

Variable	Description
3	schedule id
4	job name
5	job cpu
6	prompt number
7	prompt message
8	Schedule name
9	Schedule input arrival time (yyyymmddhhmm00)

Re-loading monitoring data

The **Configure Non-TME adapter** and **Configure TME adapter** commands set up the file **BmEvents.conf** in the *TWSHome* directory. This configuration file determines the information that the production processes (batchman and mailman) write in the *TWSHome/log_source_file* file and how this information is written. By default, this file is the *event.log* file, .

You can change the name of the log file as follows:

- Modify the **FILE** field in the *BmEvents.conf* file and restart the Tivoli Workload Scheduler processes
- Modify the **LogSource** field in the *tecad_logfile.conf* file and restarting the Tivoli Enterprise Console logfile adapter.

In the *BmEvents.conf* file the # sign represents a comment. Remove the # sign to uncomment a line.

The contents of this file are also used by other Tivoli applications that manage events, that IBM Tivoli Workload Scheduler can interact with, such as IBM Tivoli NetView and IBM Tivoli Business Systems Management.

The options you can set in the *BmEvents.conf* file are described below:

OPTIONS=MASTER | OFF

If the value is set to **MASTER** then all job scheduling events gathered by that workstation are reported. If that workstation is the master domain manager or the backup master domain manager, with Full Status option switched on, then all scheduling events for all workstations are reported.

If the value is set to **OFF**, the job scheduling events are reported only if they relate to the workstation where the file is configured.

If commented, it defaults to **MASTER** on the master domain manager workstation, and to **OFF** on a workstation other than the master domain manager.

LOGGING=ALL | KEY

Disables or enables the key flag filter mechanism.

If set to **ALL** then all events from all jobs and job streams are logged.

If set to **KEY** the event logging is enabled only for those jobs and job streams that are marked as key. The key flag is used to identify the most critical jobs or job streams. To set it in the job or job stream properties use:

- The keywords KEYSCHED (for job streams) and KEYJOB (for jobs) from the Tivoli Workload Scheduler command line interface.
- The job Monitored check box and job stream Monitored check box from the Dynamic Workload Console.

SYMEVNTS=YES|NO

If set to YES it tells the production process, **batchman**, to report the jobs and job streams status events immediately after having generated the new production day plan. This key is valid only if **LOGGING=KEY**.

If set to NO, no report is given. The default value is NO.

CHSCHED=HIGH|LOW

Indicates which events are to be sent during the job stream lifetime.

During the lifetime of a job stream its status can change several times depending on the status of the jobs it contains.

By using the **CHSCHED** option you choose how the job stream status change is reported.

If you set it to HIGH, during the job stream lifetime an event is sent any time the status of the job stream changes. Because the intermediate status of the job stream can change several times, several events can be sent, each reporting a specific status change. For example, a job stream may go into the READY state several times during its running because its status is related to the status of the jobs it contains. Each time the job stream goes into the READY state, event 153 is sent.

If you set it to LOW, during the job stream lifetime until the final status is reached, only the initial job stream state transaction is tracked. In this way the network traffic of events reporting job stream status changes is heavily reduced. When the **CHSCHED** value is set to LOW these are the events that are sent only the first time during the job stream life time:

Table 25. CHSCHED event filtered

Event number	Event Class	Description
153	TWS_Schedule_Started	Job stream started
156	TWS_Schedule_Submit	Job stream submitted
158	TWS_Schedule_Ready	Job stream ready
159	TWS_Schedule_Hold	Job stream hold
160	TWS_Schedule_Extern	Job stream external
162	TWS_Schedule	Job stream properties changed

For final status of a job stream, regardless of the value set for **CHSCHED**, all events reporting the final status of the job stream are reported, even if the job stream has more than one final status. For example, if a job contained in the job stream completes with an ABEND state, event 151 is sent (Job stream abended). If that job is then reruns and completes successfully, the job stream completes with a SUCC state and event 154 is sent (Job stream completed).

The default value for **CHSCHED** is HIGH.

EVENT=n[n ...]

Identifies which events to report in the log_source_file. Event numbers must be separated by at least one space. If omitted, the events reported by default are:

```
51 101 102 105 111 151 152 155 201 202 203 204 251 252 301
```

If the **EVENT** parameter is included, it completely overrides the defaults. To remove only event 102 from the list, for example, you must enter the following:

```
EVENT=51 101 105 111 151 152 155 201 202 203 204 251 252 301
```

Note: Event 51 is always reported each time **mailman** and **batchman** are restarted, regardless of the filters specified in the **EVENT** parameter. If you do not wish to notify this event to the TEC event console, you must manually edit the `maestro.fmt` file or, for Windows environments, the `maestro_nt.fmt` file and comment out the following section:

```
// TWS Event Log
  FORMAT TWS_Reset
  1 %s %s %s*
  event_type 1
  hostname DEFAULT
  origin DEFAULT
  agent_id $1
  software_version $2
  msg PRINTF("TWS has been reset on host %s",hostname)
  severity HARMLESS
  END
```

When this section is commented out, the TEC adapter will not send event 51 to the TEC event console.

FILE=*filename*

This option is used specifically when interacting with the Tivoli Enterprise Console. Set it to the path and file name of an ASCII log file. Job scheduling events are written to this ASCII log file which is truncated whenever the **batchman** and **mailman** processes are restarted, for example at the end of each production day.

or

FILE_NO_UTF8 =*filename*

Use this option instead of the **FILE** option when you want job scheduling events written in the local language file specified by this parameter.

Job scheduling events

After performing the configuration steps described in the “Configuring the Tivoli Enterprise Console adapter” on page 53, use the events gathered from the Tivoli Workload Scheduler log file using the Tivoli Enterprise Console logfile adapter to perform event management and correlation using the Tivoli Enterprise Console in your scheduling environment.

This section describes the events that are generated by using to the information stored in the log file specified in the `BmEvents.conf` configuration file stored on the system where you installed the Tivoli Enterprise Console logfile adapter.

An important aspect to be considered when configuring the integration with the Tivoli Enterprise Console using event adapters is whether to monitor only the master domain manager or every IBM Tivoli Workload Scheduler agent.

If you integrate only the master domain manager, all the events coming from the entire scheduling environment are reported because the log file on a master domain manager logs the information from the entire scheduling network. On the Tivoli Enterprise Console event server and TEC event console all events will therefore look as if they come from the master domain manager, regardless of which IBM Tivoli Workload Scheduler agent they originate from. The workstation name, job name, and job stream name are still reported to Tivoli Enterprise Console, but as a part of the message inside the event.

If, instead, you install a Tivoli Enterprise Console logfile adapter on every IBM Tivoli Workload Scheduler agent, this results in a duplication of events coming from the master domain manager, and from each agent. Creating and using a Tivoli Enterprise Console that detects these duplicated events, based on *job_name*, *job_cpu*, *schedule_name*, and *schedule_cpu*, and keeps just the event coming from the log file on the Tivoli Workload Scheduler agent, helps you to handle this problem. The same consideration also applies if you decide to integrate the backup master domain manager, if defined, because the log file on a backup master domain manager logs the information from the entire scheduling network. For information on creating new rules for the Tivoli Enterprise Console refer to the *IBM Tivoli Enterprise Console Rule Builder's Guide*. For information on how to define a backup master domain manager refer to *IBM Tivoli Workload Scheduler: Planning and Installation Guide*.

Figure 7 describes how an event is generated. It shows the Tivoli Enterprise Console logfile adapter installed on the master domain manager. This is to ensure that all the information about the job scheduling execution across the entire scheduling environment is available inside the log file on that workstation. You can decide, however, to install the Tivoli Enterprise Console logfile adapter on another workstation in your scheduling environment, depending on your environment and business needs.

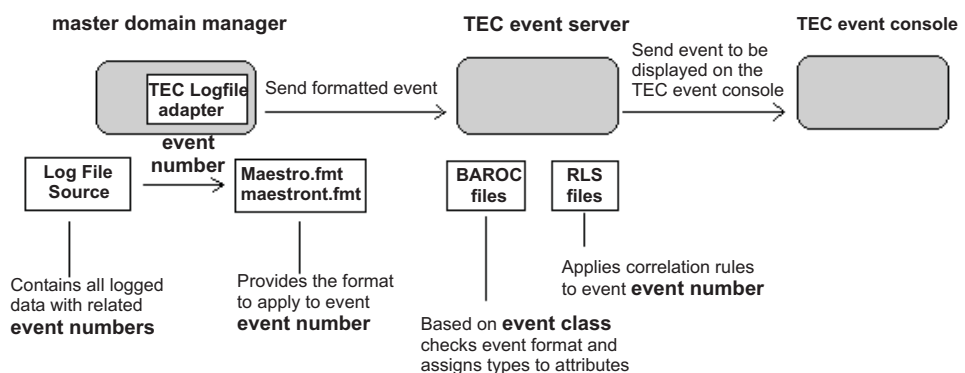


Figure 7. Event generation flow

The logic that is used to generate job scheduling events is the following:

- The information logged during the job scheduling process has an event number for each type of logged activity or problem.

- Each item of information marked with an event number that appears in the **EVENT** field of the `BmEvents.conf` file is written into the log file specified in the **FILE** field of the `BmEvents.conf` file.
- The Tivoli Enterprise Console logfile adapter reads this information inside the log file, formats it using the structure stored in the `FMT` file (`maestro.fmt` for UNIX, `maestro_nt.fmt` for Windows) and forwards it to the TEC event server, using the TEC gateway defined on the managed node of the Tivoli environment.
- On the TEC event server, the structure of the formatted information is checked using the information stored in the `BAROC` files and, if correct, is accepted. Otherwise a parsing failure is prompted.
- Once the event is accepted by the TEC event server, a check on possible predefined correlation rules or automatic responses for that event number is made using the information stored in the `RLS` files.
- If defined, the correlation rules and/or automatic responses are triggered and the event is sent to the TEC event console to be displayed on the defined Event Console.

For some error conditions on event informing that the alarm condition is ended is also stored in the log file and passed to the TEC event server via the Tivoli Enterprise Console logfile adapter. This kind of event is called a *clearing event*. It ends on the TEC event console any related problem events.

The following table describes the events and rules provided by Tivoli Workload Scheduler.

The text of the message that is assigned by the `FMT` file to the event is shown in **bold**. The text message is the one that is sent by the Tivoli Enterprise Console logfile adapter to TEC event server and then to the TEC event console. The percent sign (`%s`) in the messages indicates a variable. The name of each variable follows the message between brackets.

Table 26. Tivoli Workload Scheduler events

Event	Characteristic	Description
"TWS process %s has been reset on host %s" (<code>program_name</code> , <code>host_name</code>)	<i>Event Class:</i> <i>Event Severity:</i> <i>Event Description:</i>	TWS_Process_Reset HARMLESS Tivoli Workload Scheduler daemon process reset.
"TWS process %s is gone on host %s" (<code>program_name</code> , <code>host_name</code>)	<i>Event Class:</i> <i>Event Severity:</i> <i>Event Description:</i>	TWS_Process_Gone CRITICAL Tivoli Workload Scheduler process gone.
"TWS process %s has abended on host %s" (<code>program_name</code> , <code>host_name</code>)	<i>Event Class:</i> <i>Event Severity:</i> <i>Event Description:</i>	TWS_Process_Abend CRITICAL Tivoli Workload Scheduler process abends.

Table 26. Tivoli Workload Scheduler events (continued)

Event	Characteristic	Description
"Job %s.%s failed, no recovery specified" (schedule_name, job_name)	<p><i>Event Class:</i> TWS_Job_Abend</p> <p><i>Event Severity:</i> CRITICAL</p> <p><i>Automated Action (UNIX only):</i> Send job stdlist to the <i>TWS_user</i>.</p> <p><i>Event Description:</i> Job failed, no recovery specified.</p> <p><i>Correlation Activity:</i> If this job has abended more than once within a 24 hour time window, send a TWS_Job_Repeated_Failure event.</p>	
"Job %s.%s failed, recovery job will be run then schedule %s will be stopped" (schedule_name, job_name, schedule_name)	<p><i>Event Class:</i> TWS_Job_Abend</p> <p><i>Event Severity:</i> CRITICAL</p> <p><i>Automated Action (UNIX only):</i> Send job stdlist to the <i>TWS_user</i>.</p> <p><i>Event Description:</i> Job failed, recovery job runs, and schedule stops.</p> <p><i>Correlation Activity:</i> If this job has abended more than once within a 24 hour time window, send a TWS_Job_Repeated_Failure event.</p>	
"Job %s.%s failed, this job will be rerun" (schedule_name, job_name)	<p><i>Event Class:</i> TWS_Job_Abend</p> <p><i>Event Severity:</i> CRITICAL</p> <p><i>Automated Action (UNIX only):</i> Send job stdlist to the <i>TWS_user</i>.</p> <p><i>Event Description:</i> Job failed, the job is rerun.</p> <p><i>Correlation Activity:</i> If this job has abended more than once within a 24 hour time window, send a TWS_Job_Repeated_Failure event.</p>	
"Job %s.%s failed, this job will be rerun after the recovery job" (schedule_name, job_name)	<p><i>Event Class:</i> TWS_Job_Abend</p> <p><i>Event Severity:</i> CRITICAL</p> <p><i>Automated Action (UNIX only):</i> Send job stdlist to the <i>TWS_user</i>.</p> <p><i>Event Description:</i> Job failed, recovery job is run, and the job is run again.</p> <p><i>Correlation Activity:</i> If this job has abended more than once within a 24 hour time window, send a TWS_Job_Repeated_Failure event.</p>	
"Job %s.%s failed, continuing with schedule %s" (schedule_name, job_name, schedule_name)	<p><i>Event Class:</i> TWS_Job_Abend</p> <p><i>Event Severity:</i> CRITICAL</p> <p><i>Automated Action (UNIX only):</i> Send job stdlist to user <i>TWS_user</i>.</p> <p><i>Event Description:</i> Job failed, the schedule proceeds.</p> <p><i>Correlation Activity:</i> If this job has abended more than once within a 24 hour time window, send a TWS_Job_Repeated_Failure event.</p>	

Table 26. Tivoli Workload Scheduler events (continued)

Event	Characteristic	Description
"Job %s.%s failed, running recovery job then continuing with schedule %s" (schedule_name, job_name, schedule_name)	<p><i>Event Class:</i> TWS_Job_Abend</p> <p><i>Event Severity:</i> CRITICAL</p> <p><i>Automated Action (UNIX only):</i> Send job stdlist to the <i>TWS_user</i>.</p> <p><i>Event Description:</i> Job failed, recovery job runs, schedule proceeds.</p> <p><i>Correlation Activity:</i> If this job has abended more than once within a 24 hour time window, send a TWS_Job_Repeated_Failure event.</p>	
"Failure while rerunning failed job %s.%s" (schedule_name, job_name)	<p><i>Event Class:</i> TWS_Job_Abend</p> <p><i>Event Severity:</i> CRITICAL</p> <p><i>Automated Action (UNIX only):</i> Send job stdlist to the <i>TWS_user</i>.</p> <p><i>Event Description:</i> Rerun of abended job abends.</p> <p><i>Correlation Activity:</i> If this job has abended more than once within a 24 hour time window, send a TWS_Job_Repeated_Failure event.</p>	
"Failure while recovering job %s.%s" (schedule_name, job_name)	<p><i>Event Class:</i> TWS_Job_Abend</p> <p><i>Event Severity:</i> CRITICAL</p> <p><i>Automated Action (UNIX only):</i> Send job stdlist to the <i>TWS_user</i>.</p> <p><i>Event Description:</i> Recovery job abends.</p> <p><i>Correlation Activity:</i> If this job has abended more than once within a 24 hour time window, send a TWS_Job_Repeated_Failure event.</p>	
"Multiple failures of Job %s#%s in 24 hour period" (schedule_name, job_name)	<p><i>Event Class:</i> TWS_Job_Repeated_Failure</p> <p><i>Event Severity:</i> CRITICAL</p> <p><i>Event Description:</i> Same job fails more than once in 24 hours.</p>	
"Job %s.%s did not start" (schedule_name, job_name)	<p><i>Event Class:</i> TWS_Job_Failed</p> <p><i>Event Severity:</i> CRITICAL</p> <p><i>Event Description:</i> Job failed to start.</p>	
"Job %s.%s has started on CPU %s" (schedule_name, job_name, cpu_name)	<p><i>Event Class:</i> TWS_Job_Launched</p> <p><i>Event Severity:</i> HARMLESS</p> <p><i>Event Description:</i> Job started.</p> <p><i>Correlation Activity:</i> Clearing Event - Close open job prompt events related to this job.</p>	
"Job %s.%s has successfully completed on CPU %s" (schedule_name, job_name, cpu_name)	<p><i>Event Class:</i> TWS_Job_Done</p> <p><i>Event Severity:</i> HARMLESS</p> <p><i>Event Description:</i> Job completed successfully.</p> <p><i>Correlation Activity:</i> Clearing Event - Close open job started events for this job and auto-acknowledge this event.</p>	

Table 26. Tivoli Workload Scheduler events (continued)

Event	Characteristic	Description
"Job %s.%s suspended on CPU %s" (schedule_name, job_name, cpu_name)	Event Class: Event Severity: Event Description:	TWS_Job_Suspended WARNING Job suspended, the until time expired (default option suppress).
"Job %s.%s is late on CPU %s" (scheduler_name, job_cpu)	Event Class: Event Severity: Event Description:	TWS_Job_Late WARNING Job late, the deadline time expired before the job completed.
"Job %s.%s:until (continue) expired on CPU %s", schedule_name, job_name, job_cpu	Event Class: Event Severity: Event Description:	TWS_Job_Until_Cont WARNING Job until time expired (option continue).
"Job %s.%s:until (cancel) expired on CPU %s", schedule_name, job_name, job_cpu	Event Class: Event Severity: Event Description:	TWS_Job_Until_Canc WARNING Job until time expired (option cancel).
(TWS Prompt Message)	Event Class: Event Severity: Event Description:	TWS_Job_Recovery_Prompt WARNING Job recovery prompt issued.
"Schedule %s suspended", (schedule_name)	Event Class: Event Severity: Event Description:	TWS_Schedule_Susp WARNING Schedule suspended, the until time expired (default option suppress).
"Schedule %s is late", (schedule_name)	Event Class: Event Severity: Event Description:	TWS_Schedule_Late WARNING Schedule late, the deadline time expired before the schedule completion.
"Schedule %s until (continue) expired", (schedule_name)	Event Class: Event Severity: Event Description:	TWS_Schedule_Until_Cont WARNING Schedule until time expired (option continue).
"Schedule %s until (cancel) expired", (schedule_name)	Event Class: Event Severity: Event Description:	TWS_Schedule_Until_Canc WARNING Schedule until time expired (option cancel).
"Schedule %s has failed" (schedule_name)	Event Class: Event Severity: Event Description: Correlation Activity:	TWS_Schedule_Abend CRITICAL Schedule abends. If event is not acknowledged within 15 minutes, send mail to TWS_user (UNIX only).

Table 26. Tivoli Workload Scheduler events (continued)

Event	Characteristic	Description
"Schedule %s is stuck" (schedule_name)	<i>Event Class:</i> <i>Event Severity:</i> <i>Event Description:</i> <i>Correlation Activity:</i>	TWS_Schedule_Stuck CRITICAL Schedule stuck. If event is not acknowledged within 15 minutes, send mail to TWS_user (UNIX only).
"Schedule %s has started" (schedule_name)	<i>Event Class:</i> <i>Event Severity:</i> <i>Event Description:</i> <i>Correlation Activity:</i>	TWS_Schedule_Started HARMLESS Schedule started. Clearing Event - Close all related pending schedule, or schedule abend events related to this schedule.
"Schedule %s has completed" (schedule_name)	<i>Event Class:</i> <i>Event Severity:</i> <i>Event Description:</i> <i>Correlation Activity:</i>	TWS_Schedule_Done HARMLESS Schedule completed successfully. Clearing Event - Close all related schedule started events and auto-acknowledge this event.
(Global Prompt Message)	<i>Event Class:</i> <i>Event Severity:</i> <i>Event Description:</i>	TWS_Global_Prompt WARNING Global prompt issued.
(Schedule Prompt's Message)	<i>Event Class:</i> <i>Event Severity:</i> <i>Event Description:</i>	TWS_Schedule_Prompt WARNING Schedule prompt issued.
(Job Recovery Prompt's Message)	<i>Event Class:</i> <i>Event Severity:</i> <i>Event Description:</i>	TWS_Job_Prompt WARNING Job recovery prompt issued.
"Comm link from %s to %s unlinked for unknown reason" (hostname, to_cpu)	<i>Event Class:</i> <i>Event Severity:</i> <i>Event Description:</i>	TWS_Link_Dropped WARNING Tivoli Workload Scheduler link to CPU dropped for unknown reason.
"Comm link from %s to %s unlinked via unlink command" (hostname, to_cpu)	<i>Event Class:</i> <i>Event Severity:</i> <i>Event Description:</i>	TWS_Link_Dropped HARMLESS Tivoli Workload Scheduler link to CPU dropped by unlink command.
"Comm link from %s to %s dropped due to error" (hostname, to_cpu)	<i>Event Class:</i> <i>Event Severity:</i> <i>Event Description:</i>	TWS_Link_Dropped CRITICAL Tivoli Workload Scheduler link to CPU dropped due to error.

Table 26. Tivoli Workload Scheduler events (continued)

Event	Characteristic	Description
"Comm link from %s to %s established" (hostname, to_cpu)	<i>Event Class:</i> <i>Event Severity:</i> <i>Event Description:</i> <i>Correlation Activity:</i>	TWS_Link_Established HARMLESS Tivoli Workload Scheduler CPU link to CPU established. Close related TWS_Link_Dropped or TWS_Link_Failed events and auto-acknowledge this event.
"Comm link from %s to %s down for unknown reason" (hostname, to_cpu)	<i>Event Class:</i> <i>Event Severity:</i> <i>Event Description:</i>	TWS_Link_Failed CRITICAL Tivoli Workload Scheduler link to CPU failed for unknown reason.
"Comm link from %s to %s down due to unlink" (hostname, to_cpu)	<i>Event Class:</i> <i>Event Severity:</i> <i>Event Description:</i>	TWS_Link_Failed HARMLESS Tivoli Workload Scheduler link to CPU failed due to unlink.
"Comm link from %s to %s down due to error" (hostname, to_cpu)	<i>Event Class:</i> <i>Event Severity:</i> <i>Event Description:</i>	TWS_Link_Failed CRITICAL Tivoli Workload Scheduler CPU link to CPU failed due to error.
"Active manager % for domain %" (cpu_name, domain_name)	<i>Event Class:</i> <i>Event Severity:</i> <i>Event Description:</i>	TWS_Domain_Manager_Switch HARMLESS Tivoli Workload Scheduler domain manager switch has occurred.
Long duration for Job %s.%s on CPU %s. (schedule_name, job_name, job_cpu)	<i>Event Class:</i> <i>Event Severity:</i> <i>Event Description:</i>	TWS_Job_Launched WARNING If after a time equal to estimated duration, the job is still in exec status, a new message is generated.
Job %s.%s on CPU %s, could miss its deadline. (schedule_name, job_name, job_cpu)	<i>Event Class:</i> <i>Event Severity:</i> <i>Event Description:</i>	TWS_Job_Ready, TWS_Job_Hold WARNING If the job has a <i>deadline</i> and the sum of job estimated start time and estimated duration is greater than the deadline time, a new message is generated.
Start delay of Job %s.%s on CPU %s. (schedule_name, job_name, job_cpu)	<i>Event Class:</i> <i>Event Severity:</i> <i>Event Description:</i>	TWS_Job_Ready WARNING If the job is still in ready status, after <i>n</i> minutes a new message is generated. The default value for <i>n</i> is 10.

Default criteria that control the correlation of events and the automatic responses can be changed by editing the file `maestro_plus.rls` (in UNIX environments) or `maestront_plus.rls` (in Windows environments) file. These RLS files are created during the installation of Tivoli Workload Scheduler and compiled with the BAROC file containing the event classes for the Tivoli Workload Scheduler events

on the TEC event server when the **Setup Event Server for TWS** task is run. Before modifying either of these two files, make a backup copy of the original file and test the modified copy in your sample test environment.

For example, in the last event described in the table you can change the *n* value, the number of seconds the job has to be in ready state to trigger a new message, by modifying the rule *job_ready_open* set for the TWS_Job_Ready event class.

```
rule: job_ready_open : (
    description: 'Start a timer rule for ready',

    event: _event of_class 'TWS_Job_Ready'
    where [

        status: outside ['CLOSED'],
        schedule_name: _schedule_name,
        job_cpu: _job_cpu,
        job_name: _job_name
    ],
    reception_action: (
        set_timer(_event,600,'ready event')
    )
).
```

For example, by changing the value from 600 to 1200 in the *set_timer* predicates of the *reception_action* action, and then by recompiling and reloading the Rule Base you change from 600 to 1200 the number of seconds the job has to be in ready state to trigger a new message.

Refer to *Tivoli Enterprise Console User's Guide* and *Tivoli Enterprise Console Rule Builder's Guide* for details about rules commands.

Job scheduling events format

The integration between Tivoli Workload Scheduler and Tivoli Enterprise Console (TEC) provides the means to identify and manage a set of predefined job scheduling events. These are the events that are managed using the Tivoli Enterprise Console logfile adapter installed on the scheduling workstations. These events are listed in the following table together with the values of their positional fields. These positional fields are the ones used by the FMT files to define the event structure which, once filled up with the information stored for that specific event number in the log file, is sent by the Tivoli Enterprise Console logfile adapter to the TEC event server.

Note that since Tivoli Enterprise Console handles blank (white) spaces as delimiters, if the Tivoli Workload Scheduler events include prompts (like the one generated as CarryForward prompt), the prompts might be split in different fields in the event itself.

For additional information, refer to "Job scheduling events" on page 61.

Table 27. Event formats

Event Number	Event Class	Positional Fields Values
51	TWS_Process_Reset	<ol style="list-style-type: none"> 1. Event number. 2. Process name. 3. Local workstation name. 4. Master workstation name.

Table 27. Event formats (continued)

Event Number	Event Class	Positional Fields Values
101	TWS_Job_Abend	<ol style="list-style-type: none"> 1. Event number. 2. Job stream workstation name. 3. Job stream identified. 4. Job name. For jobs submitted with at or batch, if the name supplied by the user is not unique, this is the Tivoli Workload Scheduler-generated name, and the name supplied by the user appears as variable 8 below. 5. Workstation name on which the job runs. 6. Job number. 7. Job state, indicated by an integer: 1 (ready), 2 (hold), 3 (exec), 5 (abend), 6 (succ), 7 (concl), 8 (done), 13 (fail), 16 (intro), 23 (abenp), 24 (succp), 25 (pend). 8. Job's submitted (real) name. For jobs submitted with at or batch, this is the name supplied by the user if not unique. The unique name generated by Tivoli Workload Scheduler appears as variable 4 above. 9. Job user. 10. Name of the job's script file, or the command it runs. White space is replaced by the octal equivalent; for example, a space appears as \040. 11. The rate at which an "every" job runs, expressed as hhmm. If every was not specified for the job, this is -32768. 12. Job recovery status, indicated by an integer: 1 (stop), 2 (stop after recovery job), 3 (rerun), 4 (rerunafter recovery job), 5 (continue), 6 (continue after recovery job), 10 (this is the rerun of the job), 20 (this is the run of the recovery job). 13. An event timestamp. This is the local time on the workstation where the job event occurred. It is expressed as: yyyyymmddhhmmss00 (that is, year, month, day, hour, minute, second, hundredths always zeros). 14. Message number (not zero only for job recovery prompts). 15. The prompt number delimited by '\t', or zero if there is no prompt. 16. Job record number. Identifies in the plan the record associated to the job (not for Event number 204). 17. Job keyflag: 0 (no key flag), 1 (key flag) (not for Event number 204). 18. Effective start time of the job (not for Event number 204). It has a valid time if it occurred in the event. 19. Estimated start time of the job (not for Event number 204). It has a valid time if an Estimated Start time has been provided by the user. 20. Estimated duration of the job (not for Event number 204). Time estimated by the Tivoli Workload Scheduler engine based on statistics. 21. Deadline in Epoch (not for Event number 204). It has a valid time if a deadline time has been provided by the user. 22. The prompt text, or Tivoli Workload Scheduler error message. 23. Original schedule name (for schedules not (yet) carried forward). 24. Head job record number (different from record number for rerun?every jobs). 25. Job stream name. 26. Job stream input arrival time expressed as: yyyyymmddhhmm00.
102	TWS_Job_Failed	
103	TWS_Job_Launched	
104	TWS_Job_Done	
105	TWS_Job_Suspended	
106	TWS_Job_Submitted	
107	TWS_Job_Cancel	
108	TWS_Job_Ready	
109	TWS_Job_Hold	
110	TWS_Job_Restart	
111	TWS_Job_Failed	
112	TWS_Job_SuccP	
113	TWS_Job_Extern	
114	TWS_Job_INTRO	
115	TWS_Job_Stuck	
116	TWS_Job_Wait	
117	TWS_Job_Waitd	
118	TWS_Job_Sched	
120	TWS_Job_Late	
121	TWS_Job_Until_Cont	
122	TWS_Job_Until_Canc	
204	TWS_Job_Recovery_Prompt	

Table 27. Event formats (continued)

Event Number	Event Class	Positional Fields Values
119	TWS_Job	<ol style="list-style-type: none"> 1. Event number. 2. Job stream workstation name. 3. Job stream identifier. 4. Job name. 5. Workstation name on which the job runs. 6. Job number. 7. Property type indicated by an integer: 1 (CurrEstComplete), 2 (StartTime), 3 (StopTime), 4 (Duration), 5 (TerminatingPriority), 6 (KeyStatus). 8. Property value. 9. Record number. 10. Key flag. 11. Head job record number (different from record number for rerun?every jobs). 12. Job's submitted (real) name. For jobs submitted with at or batch, this is the name supplied by the user if not unique. The unique name generated by Tivoli Workload Scheduler appears as variable 4 above. 13. Original schedule name (for schedules not (yet) carried forward). 14. Time stamp. 15. Job stream name. 16. Job stream input arrival time expressed as: yyyyymmddhhmm00.
151	TWS_Schedule_Abend	<ol style="list-style-type: none"> 1. Event number. 2. Job stream workstation name. 3. Job stream identifier. 4. Job stream state, indicated by an integer: 1 (ready), 2 (hold), 3 (exec), 4 (stuck), 5 (abend), 6 (succ),7 (cancel). 5. Record number. 6. Key flag. 7. Original schedule name (for schedules not (yet) carried forward). 8. Time stamp. 9. Job stream name. 10. Job stream input arrival time expressed as: yyyyymmddhhmm00.
152	TWS_Schedule_Stuck	
153	TWS_Schedule_Started	
154	TWS_Schedule_Done	
155	TWS_Schedule_Susp	
156	TWS_Schedule_Submit	
157	TWS_Schedule_Cancel	
158	TWS_Schedule_Ready	
159	TWS_Schedule_Hold	
160	TWS_Schedule_Extern	
161	TWS_Schedule_CnPend	
163	TWS_Schedule_Late	
164	TWS_Schedule_Until_Cont	
165	TWS_Schedule_Until_Canc	
162	TWS_Schedule	<ol style="list-style-type: none"> 1. Event number. 2. Job stream workstation name. 3. Job stream identifier. 4. Property type indicated by an integer: 2 (StartTime), 3 (StopTime), 4 (Duration), 5. Property value. 6. Record number. 7. Original schedule name (for schedules not (yet) carried forward). 8. Time stamp. 9. Job stream name. 10. Job stream input arrival time expressed as: yyyyymmddhhmm00.

Table 27. Event formats (continued)

Event Number	Event Class	Positional Fields Values
201	TWS_Global_Prompt	<ol style="list-style-type: none"> 1. Event number. 2. Prompt name. 3. Prompt number. 4. Prompt text.
202	TWS_Schedule_Prompt	<ol style="list-style-type: none"> 1. Event number 2. Job stream workstation name. 3. Job stream identifier. 4. Job stream name. 5. Job stream input arrival time expressed as: <code>yyyymmddhhmm00</code>.
203	TWS_Job_Prompt	<ol style="list-style-type: none"> 1. Event number. 2. Job stream workstation name. 3. Job stream identifier. 4. Job name. 5. Workstation name of the job. 6. Prompt number. 7. Prompt text. 8. Job stream name. 9. Job stream input arrival time expressed as: <code>yyyymmddhhmm00</code>.
251	TWS_Link_Dropped	<ol style="list-style-type: none"> 1. Event number. 2. The "to" workstation name. 3. Link state, indicated by an integer: 1 (unknown), 2 (down due to an unlink), 3 (down due to an error), 4 (up). TWS_Link_Dropped can match link state 2 only. TWS_Link_Failed can match link state 3 only. Link states 1 and 4 do not apply to these events.
252	TWS_Link_Failed	
301	TWS_Domain_Manager_Switch	<ol style="list-style-type: none"> 1. Event number. 2. New manager. 3. The domain name. 4. Event time stamp.

Chapter 4. Integrating with Tivoli License Compliance Manager

How Tivoli Workload Scheduler integrates with Tivoli License Compliance Manager on all operating systems.

This chapter describes how Tivoli Workload Scheduler integrates with Tivoli License Compliance Manager version 2.3.

Integration with Tivoli License Compliance Manager is supported on all operating systems.

When you install Tivoli Workload Scheduler on a master domain manager, a backup master domain manager, a domain manager, a fault-tolerant agent, a standard agent, or an extended agent, depending on the operating system, the following signature files are copied to the TSW_HOME directory:

- ITWSS0805.SYS2 on Oracle
- ITWSX0805.SYS2 on AIX
- ITWSH0805.SYS2 on HP-UX
- ITWSL0805.SYS2 on Linux
- ITWSW0805.SYS2 on Windows

The new signatures are used to detect if Tivoli Workload Scheduler is installed or in use on a workstation. Before updating the Tivoli License Compliance Manager catalog with these new signatures, verify that the size of each of these files is at least of 1 KB.

Chapter 5. Integrating with Tivoli Storage Manager

How Tivoli Workload Scheduler integrates with Tivoli Storage Manager.

This chapter describes how Tivoli Workload Scheduler integrates with Tivoli Storage Manager version 5.4 and later, an enterprise-wide storage management application that provides automated storage management services to workstations, personal computers, and file servers from a variety of vendors, with a variety of operating systems.

Integration with Tivoli Storage Manager is supported only on UNIX and Linux systems.

Tivoli Storage Manager administrators perform several types of operations regularly each day using a built-in scheduling facility, which provides a simple mechanism to automate routine tasks. This scheduling facility, however, does not provide the ability to assign dependencies among scheduled tasks, to assign limits or priorities, or to define workstation classes.

Using the extended agent, you can integrate Tivoli Storage Manager with Tivoli Workload Scheduler and assign dependencies among Tivoli Storage Manager scheduled tasks or assign limits or priorities.

You can schedule the following Tivoli Storage Manager tasks using Tivoli Workload Scheduler:

- Server administrative tasks
 - Database backup
 - Volume history backup
 - Device configuration backup
 - Delete volume history
 - Inventory expiration
 - All the other administrative tasks. For details see the *IBM Tivoli Storage Manager Administrator's Reference*.

These tasks are run in the Tivoli Storage Manager Command Line Administrative Interface, which must be active.

- Client backup task

This task is run in the Tivoli Storage Manager Command Line Backup/Archive Client Interface, which must be active.

To integrate Tivoli Storage Manager with Tivoli Workload Scheduler you must set up your environment and define appropriate jobs, as described in the following sections.

Setting up the environment

To integrate Tivoli Storage Manager with Tivoli Workload Scheduler, a specific extended agent, the *tsmxagent*, is needed to run Tivoli Storage Manager commands. The *tsmxagent* runs only on UNIX and Linux systems.

When you install the Tivoli Workload Scheduler, the `tsmxagent.sh` script, representing the Tivoli Storage Manager access method, and its method options file, `tsmxagent.opts`, are copied to the `TWS_HOME/methods` directory.

To schedule a job in Tivoli Workload Scheduler to run Tivoli Storage Manager commands for administrative and client backup purposes, perform the following steps:

1. Type the user ID of the Tivoli Storage Manager administrator that accesses the Tivoli Storage Manager server in the `tsmAdmin` variable of the `tsmxagent.opts` file. The default value is `admin`.
2. Set the password of the Tivoli Storage Manager administrator, defined in the previous step, in the `TSMPASS` parameter defined using the Tivoli Workload Scheduler `parms` utility.
3. Define an extended agent workstation that uses the `tsmxagent` access method. This logical workstation definition must be hosted by a Tivoli Workload Scheduler physical workstation, either a master domain manager, a domain manager, or a fault-tolerant agent workstation. An example is the following:

```
CPUNAME TSMXAGENT
OS UNIX
NODE dontcare
TCPADDR 31111
FOR MAESTRO
HOST TWSFTA
ACCESS "tsmxagent"
TYPE X-AGENT
END
```

In this case, `TSMXAGENT` is the name of the extended agent workstation, `dontcare` is the node name, `31111` is the TCP address, `TWSFTA` is the host, and `tsmxagent` is the name of the access method referred to the `tsmxagent.sh` script. The `tsmxagent.sh` must be present in all the agents `TWS_HOME/methods` directory. The `NODE` and `TCPADDR` definitions are ignored.

Defining jobs for Tivoli Storage Manager tasks

You can schedule the following Tivoli Storage Manager tasks using Tivoli Workload Scheduler. The Tivoli Storage Manager tasks must be specified in the task string for the `tsmxagent` job.

Server administrative tasks

- Database backup (`BACKUP DB`).
- Volume history backup (`BACKUP VOLHISTORY`).
- Device configuration backup (`BACKUP DEVCONFIG`).
- Delete volume history (`DELETE VOLHISTORY`).
- Inventory expiration (`EXPIRE INVENTORY`).
- All the other administrative tasks. For details see the *IBM Tivoli Storage Manager Administrator's Reference*.

You can use Tivoli Storage Manager commands syntax for every Server administrative task. For the tasks listed above you can also use the alternative Tivoli Workload Scheduler key word listed in the following table:

Table 28. Mapping table between Tivoli Storage Manager and Tivoli Workload Scheduler administrative tasks

Tivoli Storage Manager command	Tivoli Workload Scheduler keyword
backup db	DBBACKUP
backup volhistory	VHBACKUP
backup devconfig	DCBACKUP
delete volhistory	DELVOLHIST
expire inventory	EXPIREINV
any Tivoli Storage Manager admin command	

Client backup task

- Client backup task:
 - Client backup (DEFINE SCHEDULE, DEFINE ASSOCIATION, and DEFINE CLIENTACTION).

Client backup task can be performed using different commands. The following command uses positional syntax:

- **CLIENT DOMAIN SCHEDULE CLIENT** [*optional_parms*]

where:

Table 29. Positional client backup task syntax table

Tivoli Workload Scheduler option	Description
DOMAIN	Specifies the name of the Tivoli Storage Manager policy domain to which this schedule belongs.
SCHEDULE	Specifies the name of the schedule that you want to associate with one or more clients.
CLIENTS	Specifies the name of a server-prompted Tivoli Storage Manager client ready for scheduling a backup.
OPTION ARGS	Specifies the optional parameters supported by the Tivoli Storage Manager.

The syntax of the **CLIENTACT** and **CLIENTSCHED** commands does not use positional syntax, but uses parameters to specify Tivoli Storage Manager options values, as follows:

- **CLIENTACT** -action <action name> [-domain <policy domain>] [-node <TSM node name>] [-script <value>] [-- < TSM options>]
- **CLIENTSCHED** -action <action name> -domain <policy domain> -node <TSM node name> [-polling <value>] [-script <value>] [-timeout <value>] [-schedname < TSM schedule name> | -schedprefix < TSM prefix schedule name>] [-node1] [-- < TSM options>]

where:

Table 30. Parameter client backup task syntax table

Parameter	Description	Default value
-action <action name>	Tivoli Storage Manager Define Schedule action (see <i>IBM Tivoli Storage Manager Administrator's Reference</i>).	
-domain <policy domain>	Tivoli Storage Manager Policy Domain.	

Table 30. Parameter client backup task syntax table (continued)

Parameter	Description	Default value
-node <TSM node name>	Name of the Tivoli Storage Manager client on which to run the script.	
-polling <value>	Time to wait before the next status poll.	60 seconds
-script <value>	The script to run when the -action command is specified. Use '\\\' instead of '\'. Note: Due to script name parsing, avoid using "-" in the script path\name.	
-timeout <value>	Number of polls to complete before timing out.	0 = Poll Forever
-schedname < Tivoli Storage Manager schedule name>	The name of the Tivoli Storage Manager schedule. If the length of the name is more than MAX_TSM_SUPPORTED_LENGTH , the access method will truncate the name.	TWSJOB_<JOB_ID>
-schedprefix < Tivoli Storage Manager prefix schedule name>	The prefix of the Tivoli Storage Manager schedule. The schedule name is: < TSM prefix name>_JOB_ID. If the length of the prefix is more than: MAX_TSM_SUPPORTED_LENGTH - LENGTH_JOB_ID + 1 the access method will truncate the prefix.	TWSJOB_<JOB_ID>
-node1	Specifies that, at job completion, the Tivoli Storage Manager schedule is not deleted.	By default the schedule is deleted
-- < TSM options>	Any Tivoli Storage Manager options you want to add to the Tivoli Storage Manager Define Schedule command to be run. Note: When multiple values are required for a Tivoli Storage Manager option, enclose the values in escaped double quotes, for example \"<values>\".	

The differences between the types of the client task commands are the following:

- The **CLIENT** and **CLIENTSCHED** options define a Tivoli Storage Manager schedule and associate it to the specified NODE. The Tivoli Storage Manager commands are run asynchronously. Use the **CLIENTSCHED** option when you want to trace the Tivoli Storage Manager scheduled commands which have run, because it is possible to define a unique schedule name.
- The **CLIENTACT** option allows you to launch client commands synchronously, without explicitly defining a Tivoli Storage Manager schedule. Use it when Tivoli Storage Manager is configured to automatically purge asynchronous schedules at midnight, so that you do not have to purge them manually. Tivoli Storage Manager is unable to return status information for jobs after midnight if they start before midnight.

The names of the Tivoli Storage Manager schedules must be unique to run multiple Tivoli Workload Scheduler instances of the same job at the same time. To manage the concurrency and traceability, the **tsmagent** access method has been changed. Using the **CLIENTSCHED** command, it is possible to define a unique name for the Tivoli Storage Manager schedule. There are two possibilities:

- Choose no name and allow the access method to automatically generate the Tivoli Storage Manager schedule name. This choice creates a schedule name which has the following format: "TWSJOB"<JOB_ID>, where JOB_ID is the unique identifier assigned by Tivoli Workload Scheduler to the job.
- Choose a prefix to concatenate to the <JOB_ID> to create the Tivoli Storage Manager schedule name. This choice creates a schedule name which has the following format: <schedule_prefix>_<JOB_ID>, where the <schedule_prefix> is defined by passing the new parameter -schedprefix to the **CLIENTSCHED** command and <JOB_ID> is the unique identifier assigned by Tivoli Workload Scheduler to the job.

It is also possible to choose a non-unique schedule name. In this case, concurrency and traceability are not supported. This choice is enabled passing the parameter -schedname to the **CLIENTSCHED** command. The Tivoli Storage Manager schedule is defined with the name <schedule_name>.

Using the **CLIENT** option, concurrency and traceability are not supported because, running the same Tivoli Workload Scheduler job on the x-agent, you have the following results:

- Tivoli Storage Manager schedules created with the same name are not allowed and so the Tivoli Workload Scheduler job ends unsuccessfully.
- After the unsuccessful completion of one of the Tivoli Workload Scheduler job which must run concurrently, the Tivoli Storage Manager schedule is deleted and so all the other jobs end unsuccessfully.
- The entries produced in the log files have the same name and you cannot distinguish between them.

To allow the rerun of a Tivoli Workload Scheduler job that uses the Tivoli Storage Manager extended agent, the Tivoli Storage Manager schedule must be deleted after its completion, otherwise the rerun of the Tivoli Workload Scheduler job fails because Tivoli Storage Manager recognizes that a schedule with the same name already exists. Make one of the following choices depending on which **CLIENT** option is used in the Tivoli Workload Scheduler job:

- Using the **CLIENTSCHED** option, the -node1 parameter can be used. If you pass it to the command, the job is deleted after its completion. If this argument is not passed to the command, the default behavior is that the Tivoli Storage Manager schedule is deleted. This behavior is valid both when -schedname or -schedprefix is used and when the Tivoli Storage Manager schedule name is automatically generated (no -schedname or -schedprefix are passed).
- Using the **CLIENT** option, the Tivoli Storage Manager schedule is always deleted after completion.
- Using the **CLIENTACT** option, the Tivoli Storage Manager schedule is produced automatically by Tivoli Storage Manager itself, using an internal unique name, and it is not deleted after it has run.

How to create a job

The following sections show how to create jobs for all the classes specified in the above section. The Tivoli Workload Scheduler shows standard output and return codes of the task that ran on its interfaces. Some jobs send the full job logs to the Tivoli Storage Manager command line window, so you check the Tivoli Storage Manager logs to verify that the commands ran successfully. The tsmxagent always

returns the return code that the Tivoli Storage Manager command line returns. If the return code is different from zero it is interpreted as an abend code. If you want to manage return codes different from zero you can change the x-agent jobs status using the **RCCONDSUCC** option.

Back up a database

To back up a Tivoli Storage Manager database to sequential access volumes, type the following command:

```
ADMIN DBBACKUP devclass=DEVCLASS [optional_parms]
```

where:

DEVCLASS

Specifies the name of the sequential access device class to use for the backup.

optional_parms

Specifies the optional parameters supported by the Tivoli Storage Manager.

Back up sequential volume history information

To back up sequential volume history information to one or more files, type the following command:

```
ADMIN VHBACKUP [optional_parms]
```

where:

optional_parms

Specifies the optional parameters supported by the Tivoli Storage Manager.

Back up IBM Tivoli Storage Manager device information

To back up IBM Tivoli Storage Manager device information to one or more files, type the following command:

```
ADMIN DCBACKUP [optional_parms]
```

where:

optional_parms

Specifies the optional parameters supported by the Tivoli Storage Manager.

Delete volume history file records

To delete volume history file records that are no longer needed (for example, records for obsolete database backup volumes), type the following command:

```
ADMIN DELVOLHIST type=TYPE todate=TODATE [optional_parms]
```

where:

TYPE Specifies the type of records, which also meet the date and time criteria, to delete from the volume history file. You must specify the parameters in the order shown. For details about the possible values see *IBM Tivoli Storage Manager: Administrator's Reference*.

TODATE

Specifies the date to use to select sequential volume history information to be deleted. Tivoli Storage Manager deletes only

those records with a date on or before the date you specify. You can specify the date using the following format: MM/DD/YYYY.

optional_parms

Specifies the optional parameters supported by the Tivoli Storage Manager.

Start inventory expiration processing

To start inventory expiration processing, type the following command:

```
ADMIN EXPIREINV [optional_parms]
```

where:

optional_parms

Specifies the optional parameters supported by the Tivoli Storage Manager. For details, see *IBM Tivoli Storage Manager: Administrator's Reference*.

Run any other administrating task

To run any other administrating task:

```
ADMIN [TSM_command]
```

where:

TSM_command

Specifies the command supported by the Tivoli Storage Manager. For details, see the *IBM Tivoli Storage Manager Administrator's Reference*.

Schedule a client backup

To schedule a client backup, type the following commands:

- **CLIENT DOMAIN SCHEDULE CLIENT** [*optional_parms*]

For details about optional parameters (*optional_parms*), see the *IBM Tivoli Storage Manager Administrator's Reference* guide.

- **CLIENTSCHED** -action <action name> -domain <policy domain> -node <TSM node name> [-polling <value>] [-script <value>] [-timeout <value>] [-schedname < TSM schedule name> | -schedprefix < TSM prefix schedule name>] [-node1] [-- < TSM options>]
- **CLIENTACT** -action <action name> [-domain <policy domain>] [-node <TSM node name>] [-script <value>] [-- < TSM options>]

Note: For the parameter descriptions see Table 29 on page 77 and Table 30 on page 77.

Chapter 6. Integrating with the Change and Configuration Management Database

This scenario describes what you can achieve by implementing the integration between Tivoli Workload Scheduler and dynamic workload broker with Change and Configuration Management Database. The integration facilitates the discovery and impact analysis of change management activities on the underlying system and resources on which Tivoli Workload Scheduler is installed. The administrator can perform these operations from a centralized point instead of having to combine and integrate data from independent products.

This section contains information about integrating Tivoli Workload Scheduler with Change and Configuration Management Database.

For information about integrating dynamic workload broker with Change and Configuration Management Database, see “Integrating dynamic workload broker with Configuration Management Database” on page 88.

When a new version of a product is released, or an urgent security patch must be installed, the Change and Configuration Management Database administrator must address a number of Requests For Change (RFC) on some of the workstations in the environment. She must meet a service level agreement that requires security patches to be installed in 0.5 days. To achieve this goal, the administrator uses the combined information from Change and Configuration Management Database and Tivoli Application Discovery Dependency Manager (TADDM) to obtain a detailed view of the environment and to manage changes in the state of the workstations.

She obtains information about the current workload on all the workstations affected. For example, she can see which Tivoli Workload Scheduler jobs are currently running. When the integration between Change and Configuration Management Database and Tivoli Workload Scheduler is in place, she can use the Change and Configuration Management Database GUI to select a workstation that is affected by the RFC, and then open the Tivoli Dynamic Workload Console to view and operate the Tivoli Workload Scheduler jobs running on that workstation.

From the same Change and Configuration Management Database GUI, she can also start Tivoli Provisioning Manager and start a workflow to install the patch on all the affected workstations.

Roles and skills

The following roles and skills are necessary to run the scenario:

Network Administrator

Administers the network.

Change Administrator

Administers the change management process and designs workflows to implement the change process. Required skills include Change Management Process Manager knowledge.

Configuration Administrator

Manages the configuration process management applications in Change and Configuration Management Database, including administering the

configuration management security groups and their access to applications. Required skills include Configuration Management Process Manager and TADDM knowledge.

Tivoli Workload Scheduler Administrator

Manages Tivoli Workload Scheduler workload. Required skills include Tivoli Workload Scheduler knowledge.

Tivoli Workload Scheduler Operator

Performs Tivoli Workload Scheduler operations. Required skills include Tivoli Workload Scheduler knowledge.

Tivoli Provisioning Manager Administrator

Creates and manages workflows. Required skills include Tivoli Provisioning Manager knowledge.

Hardware and software requirements

Install the following software before starting the scenario:

- Change and Configuration Management Database 7.1.1.0
- Tivoli Application Discovery Dependency Manager 7.1
- IBM Tivoli Integration Composer 7.1
- Tivoli Workload Scheduler 8.6
- Tivoli Dynamic Workload Console 8.6
- Tivoli Provisioning Manager 5.1.1

Setting up the environment

When the required software is installed, the users involved in the scenario must complete the following tasks before starting the scenario:

1. The network administrator configures the environment to support Single Sign-On between Tivoli Workload Scheduler and Tivoli Dynamic Workload Console, so that authentication is required just once for each user involved in the scenario.
2. The Configuration Administrator performs a detailed discovery of the network by using TADDM sensors. For more information about performing discoveries using TADDM, see http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/index.jsp?topic=/com.ibm.taddm.doc_7.1/cmdb_welcome.html
3. The Tivoli Workload Scheduler administrator specifies in the **localopts** file, located in the product installation directory, and the path where the IdML book must be created. The parameter **CCMDB_BOOK_PATH** is set by default to *TWS84_HOME_DIR/CCMDB*. IdML is the Discovery Library XML schema specification. IdML books, also known as Discovery Library books, are XML files containing details about resources and resource relationships written to conform to the IdML schema.
4. The Tivoli Workload Scheduler administrator exports the workstation definitions from the Tivoli Workload Scheduler database into an IdML book using the **dataextract** command on the master domain manager. No parameters are necessary for this command because all the required information is retrieved from the localopts file and useropts file. To set user credentials in the useropts file, see 'Setting user options' in the *Administration Guide*.
5. The TADDM Administrator configures the $\{\text{COLLATION_HOME}\}/\text{etc}/\text{bulkload.properties}$ file. This file gives the bulk load program the information that it needs to load the IdML book into the database.

6. The TADDM Administrator imports the IdML book into the TADDM database by using the **loadidml** command, as indicated in the example:

```
loadidml -f path_to_idml_file -h hostname -u userid -p passwd
```

For more information about the `bulkload.properties` file and the **loadidml** command, see http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/index.jsp?topic=/com.ibm.taddm.doc_7.1/cmdb_welcome.html

7. The Configuration Administrator uses IBM Tivoli Integration Composer to move the actual Tivoli Workload Scheduler CI data from TADDM to the Change and Configuration Management Database database. To enable the integration with Tivoli Workload Scheduler, the Configuration Administrator must enable the following CI types, which are specific to Tivoli Workload Scheduler:

- SYS.COMPUTERSYSTEM
- APP.TWSAGENT
- SYS.OPERATINGSYSTEM

The Configuration Administrator can also transform the CI types into actual CI types so that they can be managed directly in the RFC. For more information about importing data from TADDM, refer to *IBM Tivoli® Change and Configuration Management Database Integration Adapter for Tivoli Application Dependency Discovery Manager Implementation Guide* available at http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/index.jsp?topic=/com.ibm.ccmdb.doc_7.1/ccmdb_welcome.htm.

8. The Configuration Administrator configures the Launch in Context (LiC) parameters needed to start Tivoli Dynamic Workload Console from the Change and Configuration Management Database GUI (Maximo®). Use the LIC framework to launch the user interface for an external application with the context of the source application. Perform this operation only once, unless the parameters change.

Following the Launch In Context Guidelines, define a launch entry for all the context defined for Tivoli Workload Scheduler. When specifying a launch entry for a Tivoli Workload Scheduler action, use the following parameters for OMP product name and OMP version:

- Tivoli Workload Scheduler
- 8.6

The specific URL to be defined in the launch entry and parameters for connecting to the Tivoli Dynamic Workload Console are as follows:

```
https://WebUIHostname:https/ibm/console/xLaunch.do?pageID=com.ibm.tws.WebUI.External.navigation&showNavArea=false&action=Browse_Objs&ccmdb={sourcetoken}
```

where:

WebUIHostname

Is the fully qualified hostname or the IP address of the Tivoli Dynamic Workload Console.

https Is the port on which the Tivoli Dynamic Workload Console is listening.

Browse_Objs

Indicates the objects you want to browse in the Tivoli Dynamic Workload Console. Supported values are as follows:

BrowseJobs

Specifies that the Tivoli Dynamic Workload Console displays information about the jobs running on the selected workstation.

BrowseJobStreams

Specifies that the Tivoli Dynamic Workload Console displays information about the job streams running on the selected workstation.

{sourcetoken}

Is substituted with the value of MODELOBJECT_SOURCETOKEN taken by the attribute of the APP.TWSAGENT actual configuration item representing the Tivoli Workload Scheduler system on which you run the launch in context action.

For example, the string {source token} is changed at runtime to a value with the following syntax:

```
TWSServerHostname:TWSServerSOAPport/TWSMasterDomainName
```

The following optional parameters are also supported.

workstation

The name of the workstation to which the job stream belongs. This parameter is an alternative to the user of the {source token} keyword.

status Filters the jobs or job streams to be displayed according to their status. Supported values are:

W	Waiting
O	Successful
H	Held
R	Ready
E	Error
U	Undecided
S	Running
C	Cancelled
B	Blocked

columns

Specifies the number of columns you want to display. Supported values are:

Min	Displays a minimal set of columns
All	Displays all columns

The following is an example to show job on the system specified by {sourcetoken}:

```
https://WebUIhostname:https/ibm/console/
xLaunch.do?pageID=com.ibm.tws.WebUI.External.navigation
&showNavArea=false&action=BrowseJobs&ccmdb={sourcetoken}
```

The following is an example to show job streams on the system specified by {sourcetoken}:

```
https://WebUIhostname:https/ibm/console/
xLaunch.do?pageID=com.ibm.tws.WebUI.External.navigation
&showNavArea=false&actions=BrowseJobStreams&ccmdb={sourcetoken}
```


The following is an example, that uses the full set of options, to show jobs on a predefined Tivoli Workload Scheduler instance identified by TWSServerHostname, running on workstation TWSWorkstation, in error, undecided, running, blocked state, with all columns:

```
https://WebUIhostname:https/ibm/console/
xLaunch.do?pageID=com.ibm.tws.WebUI.External.navigation
&showNavArea=false&action=BrowseJobs&hostname=TWSServerHostname
&port=TWSServerSOAPport&workstation=TWSWorkstation&status=EUSB&columns=All
```

Using this last syntax, leave the OMP product name and the OMP version blank in the corresponding launch entry because it does not refer to a configuration item.

- The Configuration Administrator configures the Launch in Context (LiC) parameters needed to start Tivoli Provisioning Manager from the Change and Configuration Management Database GUI (Maximo). Perform this operation only once, unless the parameters change.

For more information about configuring LiC parameters for Tivoli Provisioning Manager, see <http://tioid2.torolab.ibm.com:8888/help/index.jsp>.

Figure 8 describes the integration components.

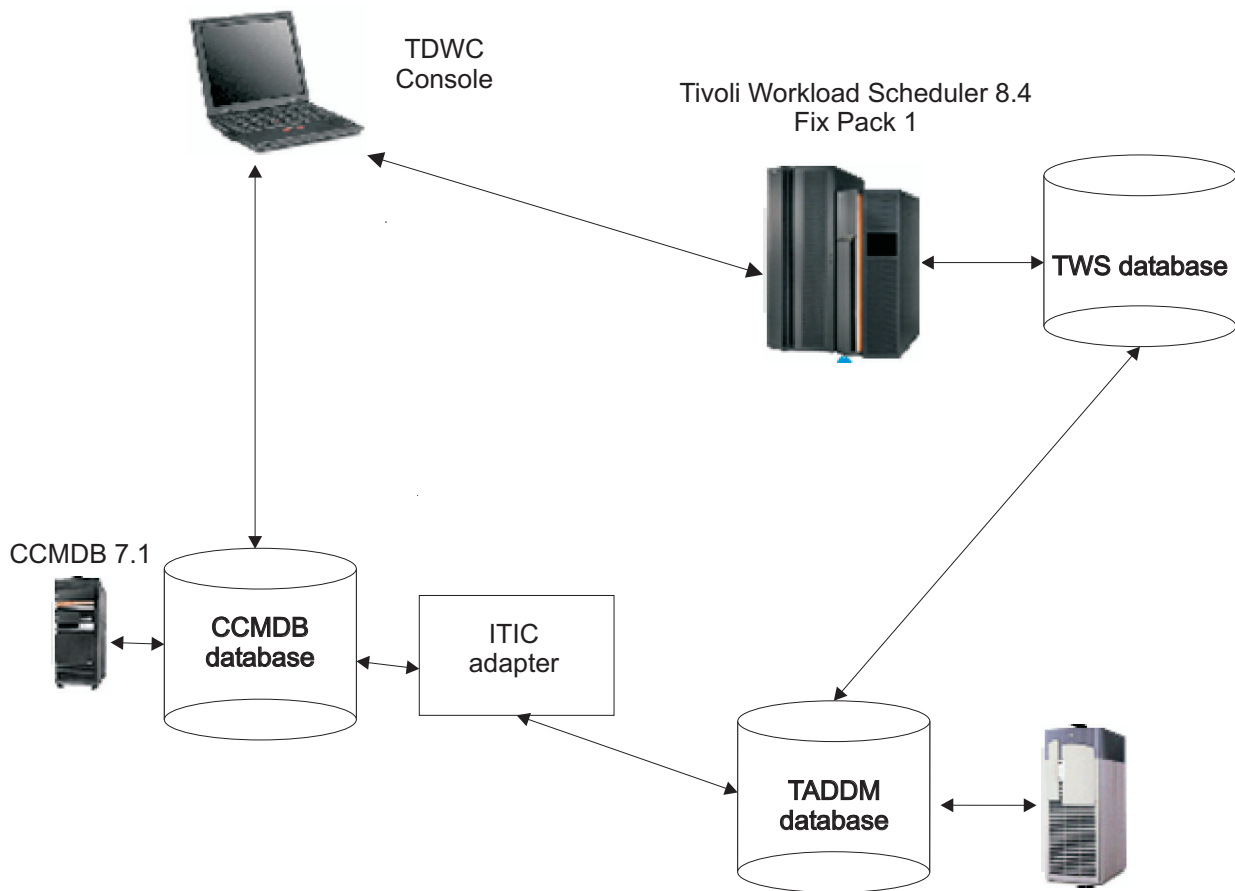


Figure 8. Integration environment

Managing RFCs using the integration between Tivoli Workload Scheduler and Change and Configuration Management Database

To complete the scenario, the users must perform the following steps:

1. After receiving the RFC, the Change Administrator verifies the state of the workstations and the installed software in Change and Configuration Management Database. For the workstations which have the Tivoli Workload Scheduler agent installed, she checks the number and status of jobs currently running by using Dynamic Workload Console.
2. The Change Administrator performs a launch in context of the Dynamic Workload Console. An instance of the Dynamic Workload Console is displayed, listing the jobs running on the selected workstation together with the related details.
3. The Change Administrator notifies the Tivoli Workload Scheduler administrator about the status of the jobs on the workstations and asks for instructions.
4. The Tivoli Workload Scheduler operator stops or pauses the jobs running on the affected workstations by using the Dynamic Workload Console.
5. The Tivoli Workload Scheduler operator notifies the Change Administrator that the jobs on the affected workstations have been stopped or paused and that she can proceed with the patch installation.
6. The Tivoli Provisioning Manager Administrator starts Tivoli Provisioning Manager.
7. The Tivoli Provisioning Manager Administrator submits a workflow that implements the RFC.
8. The Tivoli Provisioning Manager Administrator from Tivoli Provisioning Manager checks that the patch has installed successfully on all workstations.
9. The Tivoli Provisioning Manager Administrator notifies the Change administrator that the installation operations have completed on all the affected workstations.
10. The Change Administrator closes the RFC.
11. The Change Administrator notifies the Tivoli Workload Scheduler Administrator that the RFC has been closed.
12. The Tivoli Workload Scheduler operator restarts the jobs that were stopped or paused on all the affected workstations by using Dynamic Workload Console.

Integrating dynamic workload broker with Configuration Management Database

This section describes the integration of dynamic workload broker with Configuration Management Database.

Using this integration you can interact with the Tivoli Change and Configuration Management Database to manage the automatic discovery of available resources in the scheduling environment according to their characteristics and relationships.

Through the integration, dynamic workload broker imports resources from the Configuration Management Database and integrates them as resources in the environment where your jobs run. In this way you can:

- Optimize capacity across the IT infrastructure and run an enormous workload. For example, if you have a problem and a primary server cannot process the workload, you can automatically discover a new server with the same characteristics as the primary server and quickly redistribute your workload to this server. In this way you can save costs by reducing the recovery time.
- Use the resources available in the environment to support the application.
- Dynamically adapt job processing in response to changes in the environment.

- Minimize the total amount of time that is required to deliver the output of the task resolution processes.
- Assign to the job the appropriate resources it needs to run, based on the job requirements and on the administration policies.
- Optimize the use of the resources by assigning to the job the required resources based on the SLA.
- Manage and control the resource usage and load.
- Send jobs to target resources that meet the requirements needed to run the job.

The integration with Configuration Management Database is installed when you install the dynamic workload broker server. For more information about installing the Configuration Management Database enablement, see *Dynamically Scheduling your Workload*. For more information about Configuration Management Database, refer to the Configuration Management Database documentation.

The following limitations apply when importing resources:

- You can submit jobs only to endpoints that are discovered by dynamic workload broker. Any resources that are suitable to be imported must exist in the dynamic workload broker environment. This means that you can only import from Configuration Management Database the resources that run on dynamic workload broker computer systems.
- The resource-matching is performed by using the fully qualified host name of the computer on which they run. Therefore the matching can be performed only for those resources running on dynamic workload broker computer systems that have a fully qualified host name matching the host name listed in the RunsOn column in Configuration Management Database.
- You can only import resources of the AppServer type and all inherited types, because only these resource types meet the fully qualified host name criterion. You can specify the resource types to be imported in the CCMDBconfig.properties file.

A dynamic workload broker logical resource is created for each discovered Configuration Management Database resource. A relationship of the Contains type is defined between the new logical resource and the dynamic workload broker computer system defined in the RunsOn column in Configuration Management Database.

After importing the Configuration Management Database resources, they can be accessed from the dynamic workload broker Web console using the **Scheduling Environment > Logical Resources** task from the console navigation tree. They can be identified in the Logical Resources table by the value in the Owner column. The value for imported Configuration Management Database resources is CCMDB. The name of logical resources of this type cannot be modified from the dynamic workload broker Web console.

Table 31 shows details about the attribute mapping between the two products.

Table 31. Mapping between dynamic workload broker and Configuration Management Database attributes

dynamic workload broker attribute	Configuration Management Database attribute
DisplayName	Display Name or Label if Display Name is not available

Table 31. Mapping between dynamic workload broker and Configuration Management Database attributes (continued)

dynamic workload broker attribute	Configuration Management Database attribute
Name	GUID
Administrative Status	Admin State
Configuration Management Database SubType	CollationType
Creator Name	CDMSource or "CCMDB" if CDMSource is not available
Owner Name	"CCMDB"
Quantity	1

Configuring the integration with Configuration Management Database

You can use the CCMDBConfig.properties file to configure the integration with Configuration Management Database. The CCMDBConfig.properties file is installed in the following path:

installation_directory/TDWB/config

The following parameters are available in the CCMDBConfig.properties file:

CCMDB.user

Specifies the user ID used to connect to the Configuration Management Database server. You must specify this value at installation time. You can change the value afterwards. To make effective the change, you must stop and restart the dynamic workload broker server.

CCMDB.softwareElements

Specifies the comma-separated list of software elements that the user wants to import. By default, all software elements of the AppServer type are imported. You can add or change software elements in the list. If this parameter is changed, the CCMDB.lastUpdate property must be set to 0 to perform a full import instead of an update by differences. The following elements types can be specified in this parameter:

AppServer

Represents a server program. Includes the following element types: CIMOM, CitrixServer, DatabaseServer, DominoServer, GenericAppServer, J2EEServer, MQQueueManager, MySAPABAPApplicationServer, OracleAppProcessManager, ServerProcess, SMSServer, WebServer

Db2System

Represents an IBM DB2 system.

OracleServer

Represents an instance of the Oracle Server.

Service

Includes the following element types: DNSService, FileSystemService, GenericService, LDAPService, SMTPService, WebService

WebSphereNode

Represents a WebSphere® node. A node is a logical grouping of managed servers. A node usually corresponds to a physical computer system with a unique IP host address. Node names are usually identical to the host name of the computer.

Note: This information refers to the Common Data Model 2.6.5 documentation.

CMDBAddress.port

Specifies the Configuration Management Database server port. The default value is 9530. You must specify this value at installation time. You can change the value afterwards. To make effective the change, you must stop and restart the dynamic workload broker server.

CCMDB.pwd

Specifies the password for the user ID used to connect to the Configuration Management Database server. You must specify this value at installation time. The password is automatically encrypted when the command line is used. You can change the value afterwards. To make effective the change, you must stop and restart the dynamic workload broker server.

CCMDB.lastUpdate

Specifies the last time that an import operation completed successfully. The unit of measurement is milliseconds. You can modify this parameter only when the CCMDB.softwareElements parameter is changed. In this case, you must set this parameter to 0.

CMDBAddress.host

Specifies the Configuration Management Database server address. You must specify this value at installation time. You can change the value afterwards. To make effective the change, you must stop and restart the dynamic workload broker server.

The trace file is created in the following path:

installation_directory\TDWB\log\CCMDB.log

ccmdbdataimport command - Importing resources from Configuration Management Database

A dedicated command line is provided to import the resources. If you install the CCMDB enablement, the command is stored on dynamic workload broker servers in the following location:

installation_directory/TDWB/bin

Syntax

ccmdbdataimport ?

ccmdbdataimport [-usr *user_name*] [-pwd *password*] -v

Description

This command performs a number of queries on the dynamic workload broker server to retrieve computer systems on which Configuration Management Database resources can run. It then connects to the Configuration Management Database server to find the related resources.

When new resources are created or deleted in Configuration Management Database, run the command again to update the dynamic workload broker server. When an update is performed, the modifications you have applied to the resources in the dynamic workload broker environment might be overwritten with the new information from the Configuration Management Database. This is because the update operation is performed only on the resources that have changed in the Configuration Management Database rather than on all resources.

Log and trace information for this integration is stored in the dynamic workload broker command-line log and trace files. These files are stored in the following directories:

trace file

installation_directory/TDWB/logs/CLITrace.log.0

log file

installation_directory/TDWB/logs/CLI.log.0

Options

? Displays help information.

-usr *user_name*

Specifies the user name for a user who is authorized to perform operations on the command line. This parameter is required when security is enabled and the user name is not defined in the CLIconfig.properties configuration file.

-pwd *password*

Specifies the password for a user who is authorized to perform operations on the command line. This parameter is required when security is enabled and the password is not defined in the CLIconfig.properties configuration file.

-v

Used to obtain a verbose output.

Authorization

TDWBConfigurator.

Return Values

The **ccmdbdataimport** command returns one of the following values:

0 Indicates that **ccmdbdataimport** completed successfully.

< > 0 Indicates that **ccmdbdataimport** failed.

Enabling the trace for dynamic workload broker

To enable the trace for dynamic workload broker, edit the file *installation_directory*/TDWB/config/CLIconfig.properties by setting the parameter `.level` to ALL.

Chapter 7. Integrating dynamic workload broker with Tivoli Provisioning Manager

How dynamic workload broker integrates with Tivoli Provisioning Manager.

This chapter describes the integration of dynamic workload broker with Tivoli Provisioning Manager, version 7.1.1.

Tivoli Provisioning Manager is an automated resource management solution for corporate and Internet data centers. Through orchestrated provisioning, it provides the ability to manage the IT environment in real time, according to defined business policies, to achieve the desired business goals. Using this integration you can interact with Tivoli Provisioning Manager to dynamically create the resource necessary to run your workload in your IT environment. In this way you can:

- Ensure that during peak periods the jobs in the critical path are run in the required time frame.
- Automatically transfer entire workloads and update policies across multiple platforms.
- Assign to the job the appropriate resources that it needs to run, based on the job requirements and on the administration policies.
- Optimize the use of the resources by assigning to the job the required resources based on the SLA.
- Manage and control the resource usage and load.
- Dispatch jobs to target resources that meet the requirements to run the job.
- Respond to any unexpected and unplanned-for demands; individual jobs can be added ad hoc to the scheduling plan at any time.

The integration with Tivoli Provisioning Manager can be installed when installing the dynamic workload broker server. For more information about installing the Tivoli Provisioning Manager enablement, see *Tivoli Provisioning Manager Installation and Configuration Guide*.

Configuring the integration with Tivoli Provisioning Manager

The `TPMConfig.properties` file defines the integration with Tivoli Provisioning Manager and is created when the extension is installed. You can modify the file. Restart the Websphere Application Server to implement the changes.

The `TPMConfig.properties` file is installed in the following path:

`installation_directory/config`

The following parameters are available in the `TPMConfig.properties` file:

TPMAddress.hostname

Specifies the host name of the Tivoli Provisioning Manager server to be used when running the recovery action.

TPMAddress.port

Specifies the port number of the Tivoli Provisioning Manager server to be used when running the recovery action.

| **TPM.user**

| Specifies the user name of a user with the authority to run workflows.

| **TPM.pwd**

| Specifies the password for the user to be used when running a Tivoli
| Provisioning Manager workflow.

| The parameters defined in this file can be overwritten in a single job when creating
| the job with the Job Brokering Definition Console. For more information, see the
| online help for the **Scheduling** page.

| Log and trace information for this integration is stored in the Websphere
| Application Server logs and traces, stored in the following path: *WAS_profile_root*
| /AppSrv01/logs/ server1.

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 publication 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 publication. The furnishing of this publication 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.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites 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 publication 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.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

IBM, the IBM logo, and `ibm.com`[®] are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol ([®] or [™]), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Intel, and Itanium, are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Microsoft and Windows are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.



Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other company, product, and service names might be trademarks or service marks of others.

Index

A

accessibility viii

B

BmEvents.conf file
 CHSCHEM parameter 60
 configuration 59
 EVENT parameter 60
 FILE parameter 61
 FILE_NO_UTF8 parameter 61
 LOGGING parameter 59
 OPTIONS parameter 59
 SYMEVNTS parameter 60

C

CCMDB integration
 log information 92
 logging in command-line logs 92
 trace information 92
ccmdbdataimport command 91
commands
 ccmdbdataimport 91
 TEPBulkDiscovery 24
 TEPConfig 21
computer
 definition 19
Configuration Management Database
 integration
 log information 92
 logging in command-line logs 92
 trace information 92
conventions used in publications viii

D

Dynamic Workload Console
 accessibility viii

E

education viii
events
 clearing 63
 collected by adapters 61
 collected by workstations 61
 fields 69
 generation flow 62
 job scheduling events 61, 69
 log file 61
 logged 14, 60
 management 61
 modifying rules 68
 numbers 69

G

glossary viii

I

integration
 Tivoli Storage Manager 75
Integration with IBM Tivoli Monitoring
 BmEvents.conf
 EVENT parameter 15
 FILE parameter 15
 Bulk Discovery 19
 TEPConfig script 11

J

job ID
 jobcancel command 25
 jobdetails command 25
 jobstatus command 25
 retrieving 25
 TEPBulkDiscovery command 25
job scheduling events
 assigned rule 63
 automated action 63
 class 63
 clearing event 63
 correlation activity 63
 description 63
 fields 69
 format 69
 format table 69
 numbers 69
 severity 63
 syntax table 63

K

key
 keyjob 13, 60
 keysched 13, 60
 Monitored job 13, 60
 Monitored job stream 13, 60

L

logical resource
 definition 19

M

Monitored
 job 13, 60
 job stream 13, 60

P

processes
 mdemon 37
 muser 37
publications viii

S

setting
 LANG 61

T

TEC integration
 BmEvents.conf 59
 setup 61
technical training viii
TEP integration
 log information 20
 logging in WAS logs 20
 trace information 20
TEPBulkDiscovery command 24
TEPConfig command 21
Tivoli Enterprise Portal integration
 BmEvents.conf file
 CHSCHEM parameter 14
 LOGGING parameter 13
 OPTIONS parameter 13
 SYMEVNTS parameter 14
 log information 20
 logging in WAS logs 20
 trace information 20
Tivoli Provisioning Manager integration
 log information 94
 logging in WAS logs 94
 trace information 94
Tivoli Storage Manager
 integration 75
Tivoli technical training viii
TPM integration
 log information 94
 logging in WAS logs 94
 trace information 94
training
 technical viii



Product Number: 5698-WSH

Printed in USA

SC23-8904-04



Spine information:

IBM Tivoli Workload Scheduler **Version 8.6**

Integrating with Other Products

