Workload Scheduler for Applications
Version 8.6

User's Guide

IBM

SC32-1278-07

Workload Scheduler for Applications
Version 8.6

User's Guide

SC32-1278-07

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 for Applications (program number 5698-WSE) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC32-1278-06.

Contents

Part 5. z/OS access method 209

Part 6. Managing InfoSphere DataStage jobs. 235

| Part 7. Managing Cognos reports. . . 249

Figures

Tables

About this Publication

The *IBM® Tivoli® Workload Scheduler for Applications: User's Guide* provides information about how to install, set up, and use the IBM Tivoli Workload Scheduler access methods that run and control jobs of the following applications:

- Oracle E-Business Suite
- PeopleSoft
- SAP® R/3
- z/OS®

What is new in this release

For information about the new and changed functions in this release, see *Tivoli Workload Automation: Overview*.

For information about the APARs that this release addresses, see the IBM Tivoli Workload Scheduler for Applications Download Document at http://www.ibm.com/support/docview.wss?rs=672&uid=swg24032594.

Except for editorial changes, updates to this edition are marked with a vertical bar to the left of the change.

Who should read this publication

This publication is intended for job schedulers who intend to run and control jobs of the applications with IBM Tivoli Workload Scheduler. Users of this publication should have some knowledge of:

- IBM Tivoli Workload Scheduler
- Dynamic Workload Console
- The specific application environment

Publications

Full details of Tivoli Workload Scheduler publications can be found in *Tivoli Workload Automation: Publications*. This document also contains information on 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 Web site:

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:

Online
Go to the IBM Software Support site at http://www.ibm.com/software/support/probsub.html and follow the instructions.

IBM Support Assistant
The IBM Support Assistant (ISA) is a free local software serviceability workbench that helps you resolve questions and problems with IBM software products. The ISA provides quick access to support-related information and serviceability tools for problem determination. Locate and download the Tivoli Workload Scheduler for Applications Add-on and use the IBM Support Assistant interface to benefit from concurrent search, media viewer, guided troubleshooter, diagnostic tools, data collectors, service request submission, and other features. To install the ISA software, go to http://http://www-01.ibm.com/software/support/isa/.

Note: To locate and download the add-on, use the IBM Support Assistant's interface. Full instructions about how to use the application and add-on are provided within the interface. Add-ons are available at release level at the following URL: http://www.ibm.com/support/docview.wss?&uid=swg27013117).

IBM Support Assistant Lite
To quickly collect diagnostic files to solve problems faster. This is a special offering of the IBM Support Assistant that contains only the data collection component customized for a specific product. IBM Support Assistant Lite provides quick deployment of IBM Support Assistant data collection tool. It is customized to automate product-specific data collection. You can run ISA Lite to do data collection without ever installing ISA or the ISA add-on.

For more information about ISA, see the corresponding section IBM Support Assistant in Tivoli Workload Scheduler Troubleshooting Guide.

Troubleshooting Guide
For more information about resolving problems, see the problem determination information for Tivoli Workload Scheduler.

How to read syntax diagrams

Throughout this publication, syntax is described in diagrams like the one shown here, which describes the SRSTAT command:

```
►►──SRSTAT──'──resource name──'──┬──────────────────────────────────────────┬──►
                                 │                ┌─OPCA───────────┐        │
                                 └─SUBSYS──(──────┼─subsystem name─┼───)────┘
                                                  └─MSTR───────────┘
```

```
►──┬───────────────────────────┬──┬───────────────────────────────────┬──────────────►
   │           ┌─KEEP──┐       │  │                 ┌─KEEP───┐        │
   └─AVAIL──(──┼─RESET─┼──)────┘  └─DEVIATION──(────┼─amount─┼──)─────┘
               ├─NO────┤                            └─RESET──┘
               └─YES───┘

►──┬──────────────────────────────────┬──┬───────────────────────────┬───────────────►
   │              ┌─KEEP───┐          │  │             ┌─YES─┐       │
   └─QUANTITY──(──┼─amount─┼──)───────┘  └─CREATE──(───┴─NO──┴──)───┘
                  └─RESET──┘

►──┬──────────────────────────────────┬─────────────────────────────────────────────►◄
   │           ┌─0───────────┐        │
   └─TRACE──(──┴─trace level─┴──)─────┘
```

The symbols have these meanings:

```
►►──────
```

The statement begins here.

```
──────►
```

The statement is continued on the next line.

```
►──────
```

The statement is continued from a previous line.

```
──────►◄
```

The statement ends here.

Read the syntax diagrams from left to right and from top to bottom, following the path of the line.

These are the conventions used in the diagrams:

- Required items appear on the horizontal line (main path):

```
►►──STATEMENT──required item─────────────────────────────────────────────►◄
```

- Optional items appear below the main path:

```
►►──STATEMENT──┬───────────────┬─────────────────────────────────────────►◄
               └─optional item─┘
```

- An arrow returning to the left above the item indicates an item that you can repeat. If a separator is required between items, it is shown on the repeat arrow.

```
               ┌─,─────────────┐
               ▼               │
►►──STATEMENT────repeatable item─┴───────────────────────────────────────►◄
```

- If you can choose from two or more items, they appear vertically in a stack.
  - If you must choose one of the items, one item of the stack appears on the main path:

```
►►──STATEMENT──┬─required choice 1─┬─────────────────────────────────────►◄
               └─required choice 2─┘
```

- If choosing one of the items is optional, the entire stack appears below the main path:

```
►►──STATEMENT──┬───────────────────┬──────────────────────────────────►◄
               ├─optional choice 1─┤
               └─optional choice 2─┘
```

- A repeat arrow above a stack indicates that you can make more than one choice from the stacked items:

```
                 ┌─,─────────────────┐
                 ▼                   │
►►──STATEMENT────┬───────────────────┬┴───────────────────────────────►◄
                 ├─optional choice 1─┤
                 ├─optional choice 2─┤
                 └─optional choice 3─┘
```

```
                 ┌─,─────────────────┐
                 ▼                   │
►►──STATEMENT────┬─required choice 1─┬┴───────────────────────────────►◄
                 ├─required choice 2─┤
                 └─required choice 3─┘
```

- Parameters that are above the main line are default parameters:

```
             ┌─default─────┐
►►──STATEMENT─┼─────────────┼─────────────────────────────────────────►◄
             └─alternative─┘
```

- Keywords appear in uppercase (for example, STATEMENT).
- Parentheses and commas must be entered as part of the command syntax, as shown.
- For complex commands, the item attributes might not fit on one horizontal line. If that line cannot be split, the attributes appear at the bottom of the syntax diagram:

```
►►──STATEMENT──┬─required choice 1─┬──────────────┬──┬──────────────┬──┬──►◄
               │                   └─┤ option 1 ├─┘  └─┤ option 2 ├─┘  │
               ├─required choice 2─────────────────────────────────────┤
               └─required choice 3─────────────────────────────────────┘
```

option 1

```
                        ┌─default─────┐
├──optional choice 1──(──┴─alternative─┴──)──────────────────────────┤
```

option 2

```
                        ┌─default─────┐
├──optional choice 2──(──┴─alternative─┴──)──────────────────────────┤
```

Part 1. Common installation and configuration tasks

Product overview

Tivoli Workload Scheduler for Applications and IBM Tivoli Workload Scheduler for Applications Plug-ins overview.

Use access methods to extend the job scheduling capability of Tivoli Workload Scheduler to other systems and applications. Access methods run on:

- Extended agents to extend static scheduling capability.
- Dynamic agents and Tivoli Workload Scheduler for z/OS agents to extend dynamic scheduling capability.

For more details about the supported workstation that run the access methods, see "Supported agent workstations" on page 2.

Access method interact with the external system through either its command line or the Dynamic Workload Console. Tivoli Workload Scheduler for Applications includes the following access methods:

- Oracle E-Business Suite (MCMAGENT)
- PeopleSoft (psagent)
- SAP R/3 (r3batch)
- z/OS (mvsjes and mvsopc)

Using Tivoli Workload Scheduler for Applications, you can also extend the job scheduling capability with the IBM Tivoli Workload Scheduler for Applications Plug-ins to external applications. In this way, you take advantage of all Tivoli Workload Scheduler functions to manage the IBM Tivoli Workload Scheduler for Applications Plug-ins jobs.

The available plug-ins are:

- IBM InfoSphere® DataStage®
- IBMCognos®
- SAP PI Channel

To see how these elements fit together in a typical extended agent configuration and dynamic agent configuration, see Figure 1 on page 2:

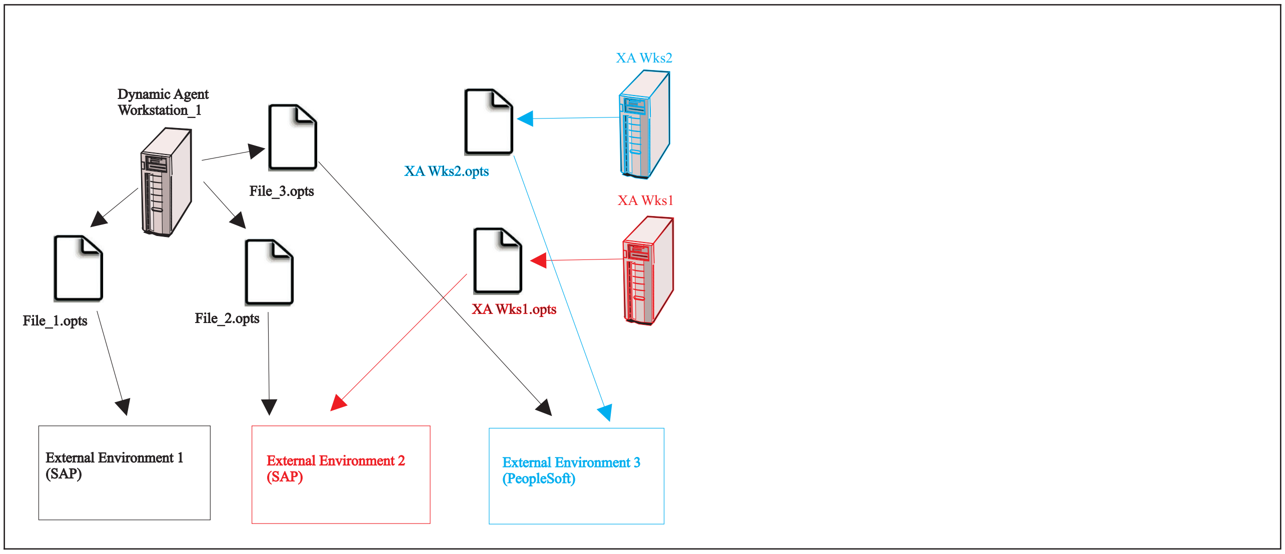


Figure 1. Supported agents configuration

Supported agent workstations

Tivoli Workload Scheduler for Applications supported agents.

You can extend IBM Tivoli Workload Scheduler capabilities for scheduling on one or more external applications, such as PeopleSoft, SAP R/3, and Oracle E-Business Suite. To extend scheduling capabilities on these external applications using Tivoli Workload Scheduler for Applications the following agents are needed:

Extended agents

Extended agents extend the static scheduling capability. They are logical workstations related to an access method hosted by a physical IBM Tivoli Workload Scheduler workstation (not another extended agent). More than one extended agent workstation can be hosted by the same Tivoli Workload Scheduler workstation and rely on the same access method. The extended agent is defined in a standard Tivoli Workload Scheduler workstation definition, which gives the extended agent a name and identifies the access method. An access method is a program that is run by the hosting workstation whenever IBM Tivoli Workload Scheduler submits a job to an external system.

Jobs are defined for an extended agent in the same manner as for other Tivoli Workload Scheduler workstations, except for any job attributes that depend on the external system or application.

To launch and monitor a job on an extended agent, the host runs the access method, passing it job details as command line options. The access method communicates with the external system to launch the job and returns the status of the job. To launch a job in an external environment, Tivoli Workload Scheduler runs the extended agent access method providing it with the extended agent workstation name and information about the job. The method looks at the corresponding file named *XANAME_accessmethod*.opts (where *XANAME* is the name of the extended agent workstation) to determine which external environment instance it connects to. The access method can then launch jobs on that instance and monitor them through completion, writing job progress and status information in the standard list file of the job.

A physical workstation can host a maximum of 255 extended agents.

Extended agents can also be used to run jobs in an end-to-end environment, where their scheduling and monitoring is performed from an IBM Tivoli Workload Scheduler for z/OS controller.

Dynamic agents and Tivoli Workload Scheduler for z/OS agents

Dynamic agents and Tivoli Workload Scheduler for z/OS agents extend the dynamic scheduling capability. To run access methods on external applications using dynamic agents, you define a job of type **Access Method**. The access method communicates with the external system to launch the job and returns the status of the job. The method uses the corresponding file named `*optionsfile_accessmethod*.opts` (where `*optionsfile*` is the configuration file that depends on the selected access method). The dynamic agent and the Tivoli Workload Scheduler for z/OS agent can have more than one associated `.opts` file to determine which external environment instance it connects to. The access method can launch jobs on that instance and monitor them through completion, writing job progress and status information in the standard list file of the job.

Installing, upgrading, and uninstalling IBM Tivoli Workload Scheduler for Applications

This chapter describes how to install, upgrade, and uninstall Tivoli Workload Scheduler for Applications and its language packs.

Planning the installation

Refer to the following sections to plan your installation of Tivoli Workload Scheduler for Applications.

Supported operating systems

For a complete list of supported operating systems, perform the following actions to produce a dynamic report:

1. Go to the web page: http://publib.boulder.ibm.com/infocenter/prodguid/v1r0/clarity/index.html.
2. Click the *"Operating systems for a specific product"* report.
3. In the windows "Operating systems for a specific product", fill in the field *Enter a full or partial product name* with the value **IBM Tivoli Workload Scheduler for Applications** and click *Search*.
4. In the *Select version* drop-down list, select the version **8.6** and click *Submit to run the report*.

For a complete list of systems requirements (disk space, temporary space and RAM usage), see http://www.ibm.com/support/docview.wss?rs=672&uid=swg27023736.

Software prerequisites

Tivoli Workload Scheduler for Applications version 8.6 prerequisites Tivoli Workload Scheduler version 8.4, or later.

Note: To define event rules based on SAP R/3 events with Tivoli Workload Scheduler for Applications version 8.6, the master domain manager must have Tivoli Workload Scheduler version 8.4 Fix Pack 1, or later, installed.

Tivoli Workload Scheduler for Applications version 8.6 interacts with the following products:

- Oracle E-Business Suite
- PeopleSoft
- SAP R/3
- z/OS

Preparing your environment

Before you install the product, ensure that:

- You have a working installation of Tivoli Workload Scheduler version 8.5.1, or later, installed on your workstation.
- You have root permissions on UNIX and administrator permissions on Windows to run the installation.
- There are no scheduling and monitoring processes currently running on the FTA where you want to perform the installation.
- For the access method for PeopleSoft, the corresponding PeopleSoft environment is already installed and fully accessible from the workstation where you are installing.
- For the access method for Oracle E-Business Suite, the Oracle E-Business Suite server is already installed on the same physical node where the access method and the Tivoli Workload Scheduler host are located.
- The installation process is not already running on the workstation. You can verify this by checking if the setup process is running.
- The Option Editor is not running.
- The X Server is installed on a workstation in your environment.
- The X-windows Display is appropriately exported if you are launching the installation process using a telnet session.
- If you are installing the product, you have the disk space and the temporary space indicated in http://www.ibm.com/support/docview.wss?rs=672&uid=swg27023736.

The installation DVD

The following list shows the installation DVDs contained in Tivoli Workload Scheduler for Applications divided by platform:

- IBM Tivoli Workload Scheduler for Applications 8.6 for AIX
- IBM Tivoli Workload Scheduler for Applications 8.6 for HP-UX for Integrity
- IBM Tivoli Workload Scheduler for Applications 8.6 for HP-UX for PA-RISC
- IBM Tivoli Workload Scheduler for Applications 8.6 for Linux
- IBM Tivoli Workload Scheduler for Applications 8.6 for Linux S390
- IBM Tivoli Workload Scheduler for Applications 8.6 for Sun Solaris
- IBM Tivoli Workload Scheduler for Applications 8.6 for Sun Solaris for Opteron
- IBM Tivoli Workload Scheduler for Applications 8.6 for Windows

Each DVD includes the following directories:

- *operating system* specific files
- INSTALLER
- LICENSE
- RESPONSE_FILE
- SAMPLES
- LAUNCHPAD
- z/OS libraries for the specific operating systems
- CLI
- GSKIT
- JVM_64

Note: On Sun Solaris for Opteron and HP-UX for Integrity, this folder is not included.

- PLUGINS
- PLUGIN_INSTALLER

Support for Internet Protocol version 6

Starting from IBM Tivoli Workload Scheduler for Applications version 8.4, Internet Protocol version 6 (IPv6) is supported in addition to legacy IPv4. Support to IPv6 is provided for:

- z/OS extended agent, with the following software requirements:
  - IBM Tivoli Workload Scheduler version 8.4, or later
  - TCP/IP for the z/OS system on IBM version 3.1, or later
- SAP extended agent, to communicate with the SAP system. IPv6 support is provided by RFC SDK version 7.10 libraries, which are supported only on the following operating systems:
  - AIX® 5.2, 5.3, 6.1 (64-bit)
  - HP-UX 11.11, 11.23, 11.31 on PA-RISC (64-bit)
  - HP-UX 11.23, 11.31 on IA64 (64-bit)
  - Linux Red Hat EL 4 on IA32 (32-bit)
  - Linux Red Hat EL 4 on IA64 (64-bit)
  - Linux Red Hat EL 4 on Power® (64-bit)
  - Linux Red Hat EL 4 on x86_64 (64-bit)
  - Linux SuSE SLES 9 on IA32 (32-bit)
  - Linux SuSE SLES 9 on IA64 (64-bit)
  - Linux SuSE SLES 9 on Power (64-bit)
  - Linux SuSE SLES 9 on x86_64 (64-bit)
  - Linux SuSE SLES 9 on zSeries® (64-bit)
  - Solaris 9, 10 on SPARC (64-bit)
  - Solaris 10 on x64 (64-bit)
  - Windows Server 2003 on IA32 (32-bit)
  - Windows Server 2003 on x64 (64-bit)
  - Windows Server 2003 on IA64 (64-bit)

Installing

The access methods and plug-ins you install with IBM Tivoli Workload Scheduler for Applications.

By installing Tivoli Workload Scheduler for Applications, you install:

- One or more access methods to interact with the following products:

  Oracle E-Business Suite
  To schedule Oracle E-Business Suite jobs by using Tivoli Workload Scheduler. For this access method, regardless of the installation procedure you choose, you must complete the installation as described in “Completing the installation of the access method for Oracle E-Business Suite” on page 15.

  PeopleSoft
  To run and monitor PeopleSoft jobs from the Tivoli Workload Scheduler environment.

SAP R/3
To create, schedule, and control SAP jobs by using the job scheduling features of Tivoli Workload Scheduler. For this access method:
- You must install the current version of SAP RFC libraries as described in SAP note 413708, before you use the SAP R/3 access method.
- If you are installing on any Windows operating system, you must also apply the SAP Note 684106 to install the required Microsoft DLLs.
- After installation, to become SAP Solution Manager Ready, see “Registering Tivoli Workload Scheduler for Applications in the SAP System Landscape Directory” on page 16.

z/OS To schedule and control z/OS jobs from the Tivoli Workload Scheduler environment.

- One or more IBM Tivoli Workload Scheduler for Applications Plug-ins to interact with the following products:

InfoSphere DataStage
To run jobs to collect data from the local database of each store and then, using the procedures stored in the central database, to produce the aggregated data g from the Tivoli Workload Scheduler environment. For more information, see “Customizing Tivoli Workload Scheduler to run InfoSphere DataStage jobs” on page 237.

Cognos
To run jobs from the Tivoli Workload Scheduler environment to create the business reports. For more information, see “Customizing Tivoli Workload Scheduler to run Cognos reports” on page 251.

SAP PI Channel
To run SAP PI Channel jobs by using the job scheduling features of Tivoli Workload Scheduler. For more information, see “Configuring external control” on page 269.

Note: For more information about the IBM Tivoli Workload Scheduler for Applications Plug-ins installation, see “Installing” on page 43.

- The Tivoli Workload Scheduler for Applications language packs related to the product you are installing.
- The Option Editor, a Java^{TM}-based file editor that you can use to configure the access methods. For details about how to use it, see “Setting options with the Option Editor” on page 26.

Note: The installation of Tivoli Workload Scheduler for Applications version 8.6.0. does not update the Software Distribution catalogs.

Throughout this guide, the path where you install Tivoli Workload Scheduler for Applications is called `TWS_home`. The default installation path depends on the Tivoli Workload Scheduler component where you install Tivoli Workload Scheduler for Applications, for example the Master Domain Manager or an agent. For detailed information about the default installation path of the Tivoli Workload Scheduler components, see the *IBM Tivoli Workload Scheduler: Planning and Installation* manual related to the product version you are using.

Before installing

Depending on the installation method you use, during the installation of Tivoli Workload Scheduler for Applications provide the following information:

Regardless of the UNZIP method you use to extract the content of eImages
The eImages of Tivoli Workload Scheduler for Applications version 8.6.0.1

are released in **zip** format. Before you install the product, extract the content of the zip files into a directory, using one of the unzip tools available on you system or downloadable from the Internet . The tool you use must be able to keep the file permissions on the extracted files, for example, infozip. You can find the unzip tool also on DVD and the eImages of the Tivoli Workload Scheduler V8.6 or V8.5.1 in the directory `TWS\<operating_system>\tws_tools`, where <operating_system> is the operating system on which you want to installed the product.

Regardless of the installation method you use
The Tivoli Workload Scheduler user that launches the access method. You can use the name of the user that you defined when you installed Tivoli Workload Scheduler. This user must belong to the Tivoli Workload Scheduler group. For example, `user1` for r3batch and `user2` for psagent.

If you are installing PeopleSoft
The full path to the `psjoa.jar` file corresponding to the release of PeopleSoft you are working with. For example, use the `psjoa.jar` file of PeopleSoft version 8.52 if you are working with version 8.52. To make the `psjoa.jar` file available on the workstation where you are installing Tivoli Workload Scheduler for Applications, perform the following steps:

1. On the workstation where the PeopleSoft application is installed, locate the *ps_home*`\class\psjoa.jar` file.
2. Either copy the file to the workstation where you are installing Tivoli Workload Scheduler for Applications, or make it accessible by mapping the path to the workstation where the PeopleSoft application is installed.

   Note: The `psjoa.jar` path cannot contain special characters.

If you are installing or upgrading SAP access method
You must first install the SAP RFC libraries. You must download the current version of these libraries, described in SAP *note 413708*, from the SAP website http://service.sap.com/swdc and copy them to the `methods` directory.

Selecting your installation method

You can install Tivoli Workload Scheduler for Applications using several different methods.

To install the product, use any of the following procedures. If you subsequently want to add another access method or install a Tivoli Workload Scheduler for Applications fix pack, ensure that you use the same installation procedure that you used initially.

Launchpad
The *launchpad* is the starting point for installing products that are part of Tivoli Workload Automation. The launchpad automatically accesses and runs the installation setup file in interactive mode. From the launchpad, you can:

- Install Tivoli Workload Scheduler for Applications or add an access method to an existing installation of Tivoli Workload Scheduler for Applications version 8.6.
- Install a Tivoli Workload Scheduler for Applications plug-in for Cognos, for InfoSphere DataStage and SAP PI Channel.
- Upgrade Tivoli Workload Scheduler for Applications version 8.3, or 8.4 to version 8.6

- Access product information

For details, see "Installing by using the launchpad."

Installation wizard
Install Tivoli Workload Scheduler for Applications or add an access method to an existing installation by running the individual setup files for each supported operating system. For details, see "Installing by using the installation wizard."

Silent installation
Customize a *response file* by adding all the configuration settings to be used during installation. Then, from the command line, run the setup command. With this procedure, you can run the installation unattended and in the background. For details, see "Installing by using the silent installation" on page 10.

Installing by using the launchpad

The launchpad is the starting point for installing products that are part of Tivoli Workload Automation. You can also use the launchpad to become familiar with the product documentation and access the following sources of information:

- Downloadable documentation
- IBM Tivoli Information Center
- IBM Publications Center

To access the launchpad, one of the following web browsers is required:

- Mozilla, version 1.7 or later
- Firefox, version 2.0 or later
- Microsoft Internet Explorer version 6.0, or later
- SeaMonkey version 1.1.4

To start the launchpad installation program:

1. Insert the DVD containing the installation image for your operating system into the drive.
2. Depending on the operating system, start the launchpad as follows:

   On Windows operating systems
   The launchpad starts automatically. If not, from the `root` directory of the DVD, run the `launchpad.exe` file.

   On UNIX and Linux operating systems
   From the `root` directory of the DVD, run the `launchpad.sh` file.

   If you want to start the launchpad from a mounted file system, ensure that you have write permission on it before starting the launchpad.
3. Read the documentation and, when you are ready to start the installation process, in the left frame of the page click **Install a specific product**.

   The Install a specific product page opens.
4. Select **Install Tivoli Workload Scheduler for Applications**.

   You are asked to insert the DVD containing the installation image for your operating system into the DVD drive. The installation program starts.

   For details about how to continue the installation, see "Installing by using the installation wizard."

Installing by using the installation wizard

To install Tivoli Workload Scheduler for Applications, perform the following steps:

1. From the product DVD for the operating system where you are installing the product, or from an eImage, run the **setup** installation command. It is located in the `*operating_system*/INSTALLER` directory. The installation starts.
2. Select the language in which you want the wizard to be displayed, and click **OK**. The Welcome panel is displayed.
3. Read the welcome information and click **Next**. The Software License Agreement panel is displayed.
4. Read the Software License Agreement information and select the radio button to accept the license agreement. Click **Next**. The Discovery instance panel is displayed with all the Tivoli Workload Automation instances found on your workstation.
5. Select the instance path where you want to install the Tivoli Workload Scheduler for Applications access methods.

   Install Tivoli Workload Scheduler for Applications 8.6
   To install one or more Tivoli Workload Scheduler for Applications access methods.

   If the installation program does not detect any Tivoli Workload Scheduler installation on the instance path you selected, you cannot perform any actions.
6. Click **Next**. The installation program lists all the access methods you can install.
7. Make as many selections as you want, and click **Next**. Depending on how many and which access methods you select, one or more additional panels request specific access method information:

   For all access methods:
   Specify the Tivoli Workload Scheduler user that runs the access method. You can use the name of the user that you defined when you installed Tivoli Workload Scheduler. This user must belong to the Tivoli Workload Scheduler group. This name is registered in the LJuser option of the option file. It is the name of the operating system user ID used to run the selected method.

   For the PeopleSoft access method:
   The fully qualified path to the `psjoa.jar` file. For detailed information, see "Before installing" on page 6.
8. Enter the requested information and click **Next**. A summary information panel is displayed.
9. Click **Next**. The installation process begins; the progress panel is displayed showing the status.
10. When the installation completes, a panel is displayed asking if you want to open the Option Editor. The Option Editor is a Java-based file editor that you use to configure the access methods you installed. For details about how to use the Option Editor, see "Setting options with the Option Editor" on page 26. If you decide not to run the Option Editor now, you can do it later.

    If you click **Yes**, the Option Editor is displayed for you to set your options for the access methods. If you click **No**, or when you have finished with the Option Editor, the installation exit panel is displayed.
11. Click **Finish** to exit the installation wizard and complete the installation.

If you received error messages, analyze the installation log files shown in Table 1 on page 10.

Table 1. Installation log files

Log file name	Content	Directory
tws4apps_status.log	Tivoli Workload Scheduler for Applications installation status log file. It reports if the installation completed successfully or with errors. In case of errors it indicates if the error is due to an incorrect field value or to a failed step.	At the begin of the installation process this log file is created in the following directory: **On Windows operating systems:** %TEMP%\twa\tws4apps **On UNIX and Linux operating systems:** $tmp\twa\tws4apps and copied to directory *Tivoli Workload Automation_installation_dir*\logs at the end of the installation process.
tws4apps_ia_install.log	Tivoli Workload Scheduler for Applications installation log file for InstallAnywhere errors.	*Tivoli Workload Automation_installation_dir*\logs
tws4apps_install.log	Tivoli Workload Scheduler for Applications installation log file.	At the begin of the installation process this log file is created in the following directory: **On Windows operating systems:** %TEMP%\twa\tws4apps **On UNIX and Linux operating systems:** $tmp\twa\tws4apps and copied to directory *Tivoli Workload Automation_installation_dir*\logs at the end of the installation process.

Installing by using the silent installation

A silent installation runs according to the parameters set in a response file. The response file includes all the installation information required to run the installation without user intervention.

To install Tivoli Workload Scheduler for Applications with the silent installation, you are provided with the following response file located under RESPONSE_FILE on the product DVD:

TWSAPPS_RespFile_<*operatingsystem*>.txt

where <*operatingsystem*> can be UNIX or Windows.

It is a template file that you can customize to reflect the type of installation you want to perform.

Using the silent installation you can install one or more access methods simultaneously. The Option Editor and language packs are also automatically installed.

When running the installation in silent mode, no messages are displayed. The messages are written in the silent installation log files listed in Table 1. If the silent installation fails, you can verify the messages written in the log files, by checking them in the section "Analyzing return codes for silent installation" on page 283.

To run the silent installation, perform the following steps:

1. Create your response file or customize the response file to include the options required to complete the installation.

   ```
   TWSAPPS_RespFile_<operatingsystem>.txt
   ```

   For a list of these options, see Table 2 on page 12.

   The response file must be accessible from the workstation where you want to perform the installation. Entries in the response file are in the format *option=value*. Each entry must be written on a separate line.
2. Insert the product DVD for your operating system and run the **setup** command, located in the *operating_system*/INSTALLER directory:

   On UNIX and Linux operating systems:

   ```
   setup.sh -i silent -f response_file
   ```

   On Windows operating systems:

   ```
   setup.bat -i silent -f response_file
   ```

   Where:

   -i silent
   Specifies that the installation is run unattended, driven by a response file.

   -f *response_file*
   Indicates the fully qualified path to the response file that contains the installation options. *response_file* can be any text file with the name and extension you choose.

Table 2 on page 12 lists the options you can specify to drive the installation.

Table 2. Options to perform a silent installation

Option	Required	Description	Value

USER_INSTALL_DIR=<*path*>	Yes	Specify the Tivoli Workload Scheduler installation path where you want to install Tivoli Workload Scheduler for Applications.	A fully qualified path. For example, to install the product under `d:\win32app\maestro`, specify:
			`USER_INSTALL_DIR= D:\win32app\maestro`
			The product files are installed in
			`D:\win32app\maestro\methods`
			On Windows operating systems: The default path is **`c:\\Program Files\\IBM\\TWA`**
			On UNIX and Linux operating systems: The default path is**`/opt/IBM/TWA`**

CHOSEN_INSTALL_FEATURE_LIST=<*value1*>,..., <*valuen*>	Yes	Specify the values that correspond to the access methods you want to install, separated by comma. Possible values are:	**R3** To install SAP R/3 **PS** To install PeopleSoft **ORA** To install Oracle E-Business Suite **ZOS** To install z/OS For example, to install the SAP R/3 and the PeopleSoft access methods, specify the following values: `CHOSEN_INSTALL_FEATURES_LIST=R3,PS`

Table 2. Options to perform a silent installation (continued)

Option	Required	Description	Value
TWSAPPS_R3_USER_NAME=<*value*>	Yes If you are installing the SAP R/3 method. That is, you specified the **R3** value in the `CHOSEN_INSTALL_FEATURES_LIST` option.	Specify the Tivoli Workload Scheduler user name to be put in the `r3batch.opts` file as **LJUser** and **IFUser**.	For example, to use the Tivoli Workload Scheduler user maestro in the `r3batch.opts` file, specify: `TWSAPPS_R3_USER_NAME=maestro` Usually you specify the name of the Tivoli Workload Scheduler user where the access method is installed.
TWSAPPS_PS_USER_NAME=<*value*>	Yes If you are installing the PeopleSoft method. That is, you specified the **PS** value in the `CHOSEN_INSTALL_FEATURES_LIST` option.	Specify the Tivoli Workload Scheduler user name to be put in the `psagent.opts` file as **LJUser** and **CFUser**.	For example, to use the Tivoli Workload Scheduler user maestro in the `psagent.opts` file, specify: `TWSAPPS_PS_USER_NAME=maestro`
TWSAPPS_PS_FILE_NAME=<*path*>	Yes If you are installing the PeopleSoft method. That is, you specified the **PS** value in the `CHOSEN_INSTALL_FEATURES_LIST` option.	The fully qualified path to the PeopleSoft `psjoa.jar` file. See "Before installing" on page 6 for detailed information.	A fully qualified path. For example: `c:\pt852\class\psjoa.jar`
TWSAPPS_ZOS_USER_NAME=<*value*>	Yes If you are installing the z/OS method. That is, you specified the **ZOS** value in the `CHOSEN_INSTALL_FEATURES_LIST` option.	Specify the Tivoli Workload Scheduler user name to be put in the `mvsjes.opts` and in the `mvsopc.opts` files as LJUser and CFUser.	For example, to use the Tivoli Workload Scheduler user maestro in the `mvsjes.opts` and in the `mvsopc.opts` files, specify: `TWSAPPS_ZOS_USER_NAME=maestro`

Table 2. Options to perform a silent installation (continued)

Option	Required	Description	Value
TWSAPPS_ORA_USER_NAME=<*value*>	Yes If you are installing the Oracle E-Business Suite method. That is, you specified the **ORA** value in the `CHOSEN_INSTALL_FEATURES_LIST` option.	Specify the Tivoli Workload Scheduler user name to be put in the `mcmagent.opts` file as LJUser.	For example, to use the Tivoli Workload Scheduler user maestro in the `mcmagent.opts` file, specify: `TWSAPPS_ORA_USER_NAME=maestro` Usually you specify the name of the Tivoli Workload Scheduler user where the access method is installed.

Note: If you are installing in silent mode, verify the status of the installation in the `tws4plugins_status.log` log file. If an error is present, check the details of the error in the `tws4plugins_install.log` log file.

The following is an example of the command you run to perform a silent installation by using the `TWSAPPS_RespFile_UNIX.txt` response file on a UNIX workstation:

```
setup.sh -i silent -f /tmp/TWSAPPS_RespFile_UNIX.txt
```

The following example shows a response file that installs the R/3 and the PeopleSoft access methods on a Windows workstation:

```
# INSTALL LOCATION
USER_INSTALL_DIR=c:\\Program Files\\IBM\\TWA
CHOSEN_INSTALL_FEATURE_LIST=R3,PS
# R3BATCH SETTINGS
TWSAPPS_R3_USER_NAME=maestro
# PSAGENT SETTINGS
TWSAPPS_PS_USER_NAME=maestro1
TWSAPPS_PS_FILE_NAME=c:\pt852\class\psjoa.jar
```

Install InfoSphere DataStage, Cognos, and SAP PI Channel plug-ins for z/OS connector on WebSphere Application Server for z/OS

Install InfoSphere DataStage, Cognos, and SAP PI Channel plug-in for z/OS connector on WebSphere® Application Server for z/OS.

To manually install InfoSphere DataStage, Cognos, and SAP PI Channel plug-ins for z/OS connector on WebSphere Application Server for z/OS, perform the following actions:

1. Ensure that you installed the **JWSZ604** FMID.
2. Locate the path where you installed the WebSphere Application Server for z/OS.
3. From the product DVD or from an eImage, copy to the `sharedLibraries` subdirectory of the WebSphere Application Server for z/OS installation directory the following files:

   For InfoSphere DataStage plug-in

   `\PLUGINS\com.ibm.scheduling.agent.datastage_8.6.0.00.jar`

For Cognos plug-in
`\PLUGINS\com.ibm.scheduling.agent.cognos_8.6.0.00.jar`

For SAP PI Channel plug-in
`\PLUGINS\com.ibm.scheduling.agent.pichannel_8.6.0.00.jar`

Completing the installation of the access method for Oracle E-Business Suite

After you have installed Tivoli Workload Scheduler for Applications, perform the following steps to complete the installation of the access method for Oracle E-Business Suite:

1. Ensure that the Oracle Applications logon has permissions to write in the `methods`, and `OracleTMP` directories located in *TWS_home*.
2. Log on as an Oracle Applications UNIX user, who has access to all the files in the Oracle Application Top directory.
3. In the *`TWS_home`*`/methods/mcmsetting.sh` file, replace the variable *Oracle_SID_database_name* with the actual name of the Oracle SID database.
4. In the Oracle Application Top directory, usually the *$APPL_TOP* installation directory, copy the content of the following files:
   - `APPS`*`contextname`*`.env` files into the `APPS`*`contextname`*`_MCM.env` file.
   - *`contextname`*`.env` files into the *`contextname`*`_MCM.env` file.

   where *contextname* is the Applications context name. The default value is the *SID_hostname*.
5. Modify the `APPS`*`contextname`*`_MCM.env` file by replacing the line:
   ```
   . /APPL_installation_path/contextname.env
   ```
   with:
   ```
   . /APPL_installation_path/contextname_MCM.env
   ```
   where *APPL_installation_path* is the complete installation path.
6. Modify the Oracle E-Business Suite *`contextname`*`_MCM.env` file as follows:
   a. At the end of the file, add the line:
   ```
   . TWS_home/methods/mcmsetting.sh
   ```
   b. Set the **APPLFENV** environment variable to the *`contextname`*`_MCM.env` file.
   c. If you are working with a version of Oracle E-Business Suite prior to 11.0, append `MCM` to the APPLSHAR and APPLFULL variables as follows:
   ```
   APPLSHAR='ALR FF MCM'
   APPLFULL='FND AD AX AK GL RG INV OP MCM'
   ```
7. Log in as the user, Oracle E-Business Suite application owner, and then run the `APPS`*`contextname`*`_MCM.env` main application environment file to set the environment:
   ```
   . APPScontextname_MCM.env
   ```
8. Make a backup copy of the *`TWS_home`*`/methods/OracleTMP` directory.
9. Change to the installed directory:
   ```
   cd TWS_home/methods/OracleTMP
   ```
10. Change the user and group ownership of the installed files to that of your Oracle E-Business Suite application owner.
11. To link the binary files, from the *`TWS_home`*`/methods/OracleTMP` directory, run the installation script **mcminstall** with the following command:
    ```
    sh mcminstall
    ```

Note: If you need to define the `TWO_TASK` environment variable to allow for an Oracle three-tier configuration, do not set it externally from **mcminstall** or in any scripts provided as input parameters. If you do, this prevents you from performing the link steps successfully.

The installation process now attempts to link the Tivoli Workload Scheduler for Applications executables with the Oracle E-Business Suite application shared libraries.

After the link process has completed successfully, you find the following executables in the `$MCM_TOP/bin` directory:

- MCMLJ
- MCMSTAT
- MCMTJ

These binary files are created by the **adrelink** utility for Oracle E-Business Suite. This utility also registers the MCM as a module in Oracle E-Business Suite.
An Oracle Tivoli shell panel is displayed.

12. Perform the following steps:
    a. Press **i** to install.
    b. Press **c** to continue the installation.
    c. Enter the name of the installation log file. The default is **mcminstall.log**, located in the *TWS_home*/`methods` directory.
    d. Enter the name of the APPS*contextname*_MCM.env main application environment file. The default is **APPLSYS.env**.
    e. Enter the *TWS_home* installation directory.
13. Edit the options file. For details, see "Configuring the Oracle E-Business Suite access method" on page 52 and "Setting options with the Option Editor" on page 26.

The installation is now complete. If you encounter problems during the link step, fix them and run the link step again, because this step is responsible for creating the binary files that enable Tivoli Workload Scheduler for Applications to interact with the Oracle Concurrent Manager when submitting and monitoring jobs.

Registering Tivoli Workload Scheduler for Applications in the SAP System Landscape Directory

Tivoli Workload Scheduler for Applications Version 8.6 is SAP Solution Manager Ready certified, with Product ID 67838200100900005452 and Software Component ID 67838200100200018803. After installing the SAP R/3 access method, register it in your SAP System Landscape Directory (SLD). The registration is valid only for the access method you register.

For detailed information about SAP Solution Manager and SAP System Landscape Directory registration of third-party software products, see SAP Note 1018839.

To register Tivoli Workload Scheduler for Applications in your SAP System Landscape Directory, perform the following steps:

1. SAP Product and Production Management System (PPMS) is the central knowledge repository for technical information regarding SAP software products and their software components. Ensure that you have a PPMS data record of IBM Tivoli Workload Scheduler for Applications in your SLD, by browsing the SLD software catalog.

If IBM Tivoli Workload Scheduler for Applications is not listed, receive an updated PPMS data collection form from SAP. For detailed information, refer to SAP Note 669669.

2. Copy the TWS4APPS860_R3BATCH_ON_*hostname*_ThirdPartySystem.xml file located in *TWS_home*/methods/solman (where *hostname* is the name of the workstation where you installed Tivoli Workload Scheduler for Applications) to the SAP workstation where the sldreg executable file is located.

   Note: On SAP 7.00 and later, sldreg is provided with the SAP kernel; on SAP NetWeaver 2004s, it is available in the SAP Service Marketplace. For detailed information about sldreg, refer to SAP Note 1018839.
3. Configure sldreg to send data to SLD by following the procedure described in the SAP online help located at the following link:

   http://help.sap.com/saphelp_nw70/helpdata/en/42/ea5ff4b5d61bd9e10000000a11466f/frameset.htm
4. Transfer the registration data to SLD by running the sldreg executable file. According to your operating system, from a command line enter:

   On Windows operating systems

   ```
   sldreg.exe -connectfile slddest.cfg
             -file TWS4APPS860_R3BATCH_ON_hostname_ThirdPartySystem.xml
   ```

   On UNIX and Linux operating systems

   ```
   sldreg -connectfile slddest.cfg
         -file TWS4APPS860_R3BATCH_ON_hostname_ThirdPartySystem.xml
   ```

   If the registration is successful, a message on the command line is returned with HTTP response code 200.

Optionally, you can verify the results of your registration in the System Landscape Directory:

1. Log on to your SLD.
2. Browse the software catalog at the **Products** link.
3. Locate the product IBM Tivoli Workload Scheduler for Applications Version 8.6 and select it.
4. Open the **Installed Systems** page and locate the workstation you registered.

Upgrading

You can upgrade Tivoli Workload Scheduler for Applications version 8.4, or 8.5 to version 8.6. By upgrading Tivoli Workload Scheduler for Applications, you upgrade:

- All the access methods you have installed are automatically upgraded to version 8.6.0.

  If you are upgrading the SAP R/3 access method on any Windows operating system, you must also apply the SAP Note 684106 to install the required Microsoft DLLs.
- The upgrade of Tivoli Workload Scheduler for Applications version 8.4.0 or 8.5.0 to version 8.6.0. does not update the Software Distribution catalogs.
- All the Tivoli Workload Scheduler for Applications language packs. Even if in the previous installation you did not install any language pack, all language packs are automatically installed when you upgrade to version 8.6.

Upgrading using the InstallAnywhere installation wizard

To upgrade Tivoli Workload Scheduler for Applications using the InstallAnywhere installation wizard, perform the following steps:

1. From the product DVD, or from an eImage, run the **setup** installation command for the operating system where you are installing the product. The **setup** installation commands are located in the `*operating_system*/INSTALLER` directory. The installation starts.
2. Select the language in which you want the wizard to be displayed, and click **OK**. The Welcome panel is displayed.
3. Read the welcome information and click **Next**. The Software License Agreement panel is displayed.
4. Read the Software License Agreement information and select the radio button to accept the license agreement. Click **Next**.
5. The installation program lists all the Tivoli Workload Automation instances found on your workstation. From the list, select the instance path that you want to upgrade. The bottom frame of the panel displays the instance content and the action you can perform:

   Upgrade to Tivoli Workload Scheduler for Applications 8.6
   To upgrade **all** Tivoli Workload Scheduler for Applicationsversion 8.4, or 8.5 access method to version 8.6.0.

   Install Tivoli Workload Scheduler for Applications 8.6
   To install the new access method.
6. Click **Next**. A summary information window is displayed.
7. Click **Next**. The installation process begins the upgrade process. The progress window is displayed showing the status of the upgrade.
8. When the upgrade has completed, a window is displayed asking if you want to open the Option Editor.

   The Option Editor is a Java-based file editor that you use to define options for the access methods you installed. If you decide not to run the Option Editor now, you can do it later. If you click **Yes**, the Option Editor is displayed and you can set your options for the methods you installed. See "Setting options with the Option Editor" on page 26 to learn how to use the Option Editor.

   If you click **No**, or when you have finished with the Option Editor, the installation exit window is displayed.
9. Click **Finish** to exit the installation wizard and complete the upgrade. If you received error messages, analyze the log files shown in Table 3.

Table 3. Upgrade wizard log files

Log file name	Content	Directory
`tws4apps_status.log`	Tivoli Workload Scheduler for Applications status log file. It reports if the upgrade completed successfully or with errors. In case of errors it indicates if the error is due to an incorrect field value or to a failed step.	*Tivoli Workload Automation_installation_dir*\logs
`tws4apps_ia_install.log`	Tivoli Workload Scheduler for Applications log file for InstallAnywhere errors.	*Tivoli Workload Automation_installation_dir*\logs
`tws4apps_install.log`	Tivoli Workload Scheduler for Applications log file.	*Tivoli Workload Automation_installation_dir*\logs

Upgrading using the silent installation

During upgrade installation process you can:

Upgrade to Tivoli Workload Scheduler for Applications 8.6

To upgrade **all** Tivoli Workload Scheduler for Applicationsversion 8.4, or 8.5 access method to version 8.6.0.

Install Tivoli Workload Scheduler for Applications 8.6 or IBM Tivoli Workload Scheduler for Applications Plug-ins 8.6

To install the new access method or the plug-ins.

To upgrade Tivoli Workload Scheduler for Applications using the silent installation, you must follow the same procedure outlined in "Installing by using the silent installation" on page 10. To install the IBM Tivoli Workload Scheduler for Applications Plug-ins during upgrade phase you need the options related to them, see Table 4. The following table Table 4 shows the options required for a silent upgrade.

Table 4. Options to perform a silent installation

Option	Required	Description	Value

USER_INSTALL_DIR=*<path>*	Yes	Specify the Tivoli Workload Scheduler installation path where you want to install Tivoli Workload Scheduler for Applications.	A fully qualified path. For example, to install the product under `d:\win32app\maestro`, specify:
			`USER_INSTALL_DIR= D:\win32app\maestro`
			The product files are installed in
			`D:\win32app\maestro \methods`
			On Windows operating systems: The default path is **`c:\\Program Files\\IBM\\ TWA`**
			On UNIX and Linux operating systems: The default path is**`/opt/IBM/ TWA`**

Table 4. Options to perform a silent installation (continued)

Option	Required	Description	Value
CHOSEN_INSTALL_FEATURE_LIST=<*value1*>,..., <*valuen*>	Yes	Specify the values that correspond to the access methods you want to install or upgrade, separated by comma. **Note: You must specify all the access methods you have installed.**Possible values are:	**R3** To install SAP R/3 **PS** To install PeopleSoft **ORA** To install Oracle E-Business Suite **ZOS** To install z/OS **Note:** For IBM Tivoli Workload Scheduler for Applications Plug-ins options, see Table 4 on page 19.For example, to install the SAP R/3 and the PeopleSoft access methods, specify the following values: `CHOSEN_INSTALL_FEATURES_LIST=R3,PS`
TWSAPPS_R3_USER_NAME=<*value*>	Yes If you are installing the SAP R/3 method. That is, you specified the **R3** value in the `CHOSEN_INSTALL_FEATURES_LIST` option.	Specify the Tivoli Workload Scheduler user name to be put in the `r3batch.opts` file as **LJUser** and **IFUser**.	For example, to use the Tivoli Workload Scheduler user maestro in the `r3batch.opts` file, specify: `TWSAPPS_R3_USER_NAME=maestro` Usually you specify the name of the Tivoli Workload Scheduler user where the access method is installed.
TWSAPPS_PS_USER_NAME=<*value*>	Yes If you are installing the PeopleSoft method. That is, you specified the **PS** value in the `CHOSEN_INSTALL_FEATURES_LIST` option.	Specify the Tivoli Workload Scheduler user name to be put in the `psagent.opts` file as **LJUser** and **CFUser**.	For example, to use the Tivoli Workload Scheduler user maestro in the `psagent.opts` file, specify: `TWSAPPS_PS_USER_NAME=maestro`
TWSAPPS_PS_FILE_NAME=<*path*>	Yes If you are installing the PeopleSoft method. That is, you specified the **PS** value in the `CHOSEN_INSTALL_FEATURES_LIST` option.	The fully qualified path to the PeopleSoft `psjoa.jar` file. See "Before installing" on page 6 for detailed information.	A fully qualified path. For example: `c:\pt852\class\psjoa.jar`

Table 4. Options to perform a silent installation (continued)

Option	Required	Description	Value
TWSAPPS_ZOS_USER_NAME=<*value*>	Yes If you are installing the z/OS method. That is, you specified the **ZOS** value in the `CHOSEN_INSTALL_FEATURES_LIST` option.	Specify the Tivoli Workload Scheduler user name to be put in the `mvsjes.opts` and in the `mvsopc.opts` files as LJUser and CFUser.	For example, to use the Tivoli Workload Scheduler user maestro in the `mvsjes.opts` and in the `mvsopc.opts` files, specify: `TWSAPPS_ZOS_USER_NAME=maestro`
TWSAPPS_ORA_USER_NAME=<*value*>	Yes If you are installing the Oracle E-Business Suite method. That is, you specified the **ORA** value in the `CHOSEN_INSTALL_FEATURES_LIST` option.	Specify the Tivoli Workload Scheduler user name to be put in the `mcmagent.opts` file as LJUser.	For example, to use the Tivoli Workload Scheduler user maestro in the `mcmagent.opts` file, specify: `TWSAPPS_ORA_USER_NAME=maestro` Usually you specify the name of the Tivoli Workload Scheduler user where the access method is installed.

If the silent upgrade fails, you can verify the messages written in the log files, by checking them in the section“Analyzing return codes for silent installation” on page 283

Note: All the access methods you have installed are automatically upgraded to version 8.6.0.

Uninstalling

Uninstalling the product either using the wizard or the silent method.

When you uninstall Tivoli Workload Scheduler for Applications, you can uninstall one or more access methods simultaneously. If you uninstall all the installed access methods, the uninstallation program removes also the Option Editor and all the language packs. For Oracle E-Business Suite, regardless of the uninstallation procedure you choose, you must complete the uninstallation as described in “Completing the removal of the access method for Oracle E-Business Suite” on page 25.

If you installed Tivoli Workload Scheduler for Applications by using the silent installation, uninstall it using the same procedure.

The uninstall program does not delete the files that you modified and the installation log files. To delete them specify the **-DREMOVE_ALL_FILES=true** option when uninstalling the product.

To uninstall Tivoli Workload Scheduler for Applications, you can use any of the following procedures:

Wizard
For details, see “Uninstalling by using the wizard.”

Silent For details, see “Uninstalling by using the silent uninstallation” on page 23.

Before uninstalling

Before you uninstall the product, ensure that the following conditions are satisfied:

- You have the following permissions to run the installation:
  Windows operating systems:
  Administrator
  UNIX and Linux operating systems:
  Root
- There are no scheduling and monitoring processes currently running on the agent where you want to uninstall.
- The installation process is not already running on the workstation. You can verify it by checking that the **setup** processes is not running.
- The Option Editor is not running.

Uninstalling by using the wizard

Uninstalling Tivoli Workload Scheduler for Applications by using the wizard and deleting also the files you modified and the log files.

Using the wizard, you can uninstall the entire product or one access method at a time. All the language packs and the Option Editor are uninstalled simultaneously when you remove the last access method.

If you installed the product using the installation wizard, you can uninstall it either using the uninstallation wizard or the silent uninstallation. If you installed the product using the silent installation you must use the silent uninstallation to uninstall it.

The uninstall program does not delete the files that you modified and the installation log files. To delete them specify the **-DREMOVE_ALL_FILES=true** option when uninstalling the product.

To uninstall the product, perform the following steps:

1. Run the following command:

   To uninstall the product without deleting the files that you modified and the log files:

   On Windows operating systems:
   From the *TWS_home*\methods directory,
   ```
   .\_tws4apps_uninst\_tws4apps_uninst.exe
   ```
   On UNIX and Linux operating systems:
   From the *TWS_home*/methods directory,
   ```
   ./_tws4apps_uninst/_tws4apps_uninst
   ```

   To uninstall the product deleting the files that you modified and the installation log files:

   On Windows operating systems:
   From the *TWS_home*\methods directory,
   ```
   .\_tws4apps_uninst\_tws4apps_uninst.exe -DREMOVE_ALL_FILES=true
   ```
   On UNIX and Linux operating systems:
   From the *TWS_home*/methods directory,
   ```
   ./_tws4apps_uninst/_tws4apps_uninst -DREMOVE_ALL_FILES=true
   ```

The uninstall program starts.

2. Select the language for the wizard, and click **OK**. The Welcome window is displayed.
3. Read the welcome information and click **Next**.
4. A summary information window is displayed listing all the access methods that you can uninstall. By default the product uninstalls all the access methods.
5. Clear the access methods you do not want to uninstall and click **Next**. The uninstall process starts. When the uninstallation completes, a panel showing the results is displayed.
6. Click **Finish** to exit the uninstallation wizard.

   Note: If you run the **uninstaller** command from the `_tws4apps_uninst` directory, you must exit from the `_tws4apps_uninst` directory to complete the uninstallation. After the uninstallation completed successfully, remove the `_tws4apps_uninst` directory.

If you received error messages, analyze the uninstallation log files listed in Table 5.

Table 5. Uninstallation log files

Log file name	Content	Directory
`tws4apps_status.log`	Tivoli Workload Scheduler for Applications installation status log file. It reports if the uninstallation completed successfully or with errors. In case of errors it indicates if the error is due to an incorrect field value or to a failed step.	*Tivoli Workload Automation_installation_dir*\logs
`tws4apps_ia_uninstall.log`	Tivoli Workload Scheduler for Applications log file for InstallAnywhere errors.	*Tivoli Workload Automation_installation_dir*\logs
`tws4apps_uninstall.log`	Tivoli Workload Scheduler for Applications log file.	*Tivoli Workload Automation_installation_dir*\logs

Uninstalling by using the silent uninstallation

Use the silent uninstallation to uninstall the product in silent mode, without user intervention. Using the silent installation you can uninstall all the installed access methods simultaneously or one access method at a time. Moreover the uninstall program gives you also the possibility to delete the files you modified and the installation log files.

If you installed the product using the installation wizard, you can uninstall it either using the uninstallation wizard or the silent uninstallation. If you installed the product using the silent installation you must use the silent uninstallation to uninstall it.

When running the uninstallation in silent mode, no messages are displayed. The messages are written in the silent installation log files listed in Table 5. If the silent uninstallation fails, you can verify the messages written in the log files, by checking them in the section“Analyzing return codes for silent installation” on page 283.

The uninstall program does not delete the files that you modified and the installation log files. To delete them specify the **-DREMOVE_ALL_FILES=true** option when uninstalling the product.

Depending on what you want to uninstall, run one of the following commands:

To uninstall one or more access method at a time, without deleting the files that you modified and the log files, run the following command:

On Windows operating systems:

From the *TWS_home*\methods directory,

```
tws4apps_uninst.exe -i silent -f <response file>
```

Where *response file* is a template file that you can customize to indicate the list of access method you want to uninstall. The default response file is `TWSAPPS_RespFile_Uninst_windows.txt`. It is located under the `RESPONSE_FILE` directory.

On UNIX and Linux operating systems:

From the *TWS_home*/methods directory,

```
./_tws4apps_uninst/_tws4apps_uninst -i silent -f <response file>
```

Where *response file* is a template file that you can customize to indicate the list of access method you want to uninstall. The default response file is `TWSAPPS_RespFile_Uninst_unix.txt`. It is located under the `RESPONSE_FILE` directory

To uninstall one or more access method at a time, deleting the files that you modified and the log files, run the following command:

On Windows operating systems:

From the *TWS_home*\methods directory,

```
tws4apps_uninst -i silent -f <response file> -DREMOVE_ALL_FILES=true
```

Where *response file* is a template file that you can customize to indicate the list of access method you want to uninstall. The default response file is `TWSAPPS_RespFile_Uninst_windows.txt`. It is located under the `RESPONSE_FILE` directory.

On UNIX and Linux operating systems:

From the *TWS_home*/methods directory,

```
./_tws4apps_uninst/_tws4apps_uninst.bin -i silent
-f <response file> -DREMOVE_ALL_FILES=true
```

Where *response file* is a template file that you can customize to indicate the list of access method you want to uninstall. The default response file is `TWSAPPS_RespFile_Uninst_unix.txt`. It is located under the `RESPONSE_FILE` directory.

Note: If you want uninstall also the IBM Tivoli Workload Scheduler for Applications Plug-ins, see Table 11 on page 50.

Table 6 on page 25 lists the options you can specify when uninstalling.

Table 6. Options to perform a silent uninstallation

Option	Required	Description	Value
FEATURE_UNINSTALL_LIST=<*value*>	Yes	Specify the values that correspond to the access methods you want to uninstall, separated by comma.	Possible values are: **R3 Batch Method** To uninstall SAP R/3 **PeopleSoft Method** To uninstall PeopleSoft **Oracle Method** To uninstall Oracle E-Business Suite **ZOS Method** To uninstall z/OS For example, to install the SAP R/3 and the PeopleSoft access methods, specify the following values: `FEATURE_UNINSTALL_LIST= R3 Batch Method, PeopleSoft Method`

The following is an example of the command you run from the directory *TWS_home*\methods, to perform a silent installation on a UNIX workstation, by using the response file TWSAPPS_RespFile_UNIX.txt:

```
./_tws4apps_uninst/_tws4apps_uninst.bin -i silent -f TWSAPPS_RespFile_UNIX.txt
```

The following example shows a response file that uninstalls the SAP R/3 and the PeopleSoft access methods:

```
FEATURE_UNINSTALL_LIST=R3 Batch Method,PeopleSoft Method
```

Completing the removal of the access method for Oracle E-Business Suite

For the Oracle E-Business Suite access method, when the uninstallation process has completed, perform the following steps:

1. Restore APPLSYS.env to its original contents by removing the MCM string from the APPLSHAR and APPLFUL variables. For example:

   ```
   APPLSHAR='ALR FF MCM'
   ```

   becomes:

   ```
   APPLSHAR='ALR FF'
   ```

2. Remove the following files:
   - *TWS_home*/methods/MCMAGENT
   - *TWS_home*/catalog/C/mcm.cat
   - MCMLJ, MCMSTAT, and MCMTJ located in the $MCM_TOP/bin directory
   - MCMLJ.O, MCMSTAT.O, and MCMTJ.O located in the $MCM_TOP/LIB directory
   - MCMprod.txt located in the $APPL_TOP/admin directory
   - All the options files

Configuring IBM Tivoli Workload Scheduler for Applications

To use supported agents, you create an options file, which configures the access method and defines the workstation and the jobs that extend the scheduling capability to external systems or applications.

Setting options with the Option Editor

An options file is a text file located in the `methods` directory of the Tivoli Workload Scheduler for Applications installation, containing a set of options to customize the behavior of the access method. The options must be written one per line and have the following format (with no spaces included):

`option=value`

Extended agents

All access methods use two types of options file:

Global options file

A common configuration file created by default for each access method installed, whose settings apply to all the extended agent workstations defined for that method. When the global options file is created, it contains only the **LJuser** option, which represents the operating system user ID used to launch the access method. You can customize the global options file by adding the options appropriate to the access method.

The name of the global options file is *`accessmethod`*`.opts`, which, depending on your operating system, corresponds to:

For Oracle E-Business Suite
`MCMAGENT.opts`

For PeopleSoft
`psagent.opts`

For SAP R/3
`r3batch.opts`

For z/OS
`mvsjes.opts`, `mvsopc.opts`

Local options file

A configuration file that is specific to each extended agent workstation within a particular installation of an access method. The name of this file is *`XANAME_accessmethod`*`.opts`, where:

XANAME
Is the name of the extended agent workstation.

accessmethod
Is the name of the access method.

If you do not create a local options file, the global options file is used. Every extended agent workstation, except for z/OS, must have a local options file with its own configuration options.

Note: The value for *XANAME* must be written in uppercase alphanumeric characters. Double-byte character set (DBCS), single-byte character set (SBCS), and bidirectional text are not supported. For information about acceptable values for the extended agent workstation name, See Table 7 on page 32.

For example, if the installation of the `r3batch` access method includes two extended agent workstations, `CPU1` and `CPU2`, the names of the local options files are `CPU1_r3batch.opts` and `CPU2_r3batch.opts`.

Dynamic agents and Tivoli Workload Scheduler for z/OS agents

All access methods use two types of options files:

Global options file

A common configuration file created by default for each access method installed, whose settings apply to all the dynamic agent workstations defined for that method. When the global options file is created, it contains only the **LJuser** option, which represents the operating system user ID used to launch the access method. You can customize the global options file by adding the options appropriate to the access method.

The name of the global options file is `*accessmethod*.opts`, which, depending on your operating system, corresponds to:

For Oracle E-Business Suite

`MCMAGENT.opts`

For PeopleSoft

`psagent.opts`

For SAP R/3

`r3batch.opts`

For z/OS

`mvsjes.opts`, `mvsopc.opts`

Local options file

One or more configuration files that are specific to each access method. The name of this file is *optionsfile_accessmethod*`.opts`.

In a distributed environment

- If you are defining a job to run the access method by using the Dynamic Workload Console, it is the option file you specify in the **New** > **Job definition** > **ERP** > **Access Method** XA Task tab.
- If you are defining the SAP R/3 job to run the access method by using the Dynamic Workload Console, it is the option file you specify in the **New** > **Job definition** > **ERP** > **SAP Job on Dynamic Workstations** XA Task tab.
- If you are defining the job to run the access method by using the **composer**, it is the option file you specify in the **target** attribute of the job definition.

If you do not create a local options file, the global options file is used.

In a z/OS environment

- If you are defining a job to run the access method by using the Dynamic Workload Console, it is the option file you specify in the **New** > **ERP** > **Access Method** XA Task tab.
- If you are defining the SAP R/3 job to run the access method by using the Dynamic Workload Console, it is the option file you specify in the **New** > **ERP** > **SAP** XA Task tab.
- If you are defining the job to run the access method by using the **JOBREC** statement, it is the name of the workstation where the access method runs.

If you do not create a local options file, the global options file is used.

If you do not specify an option in the *optionsfile_accessmethod*.opts file, while the access method is running, the product uses the values specified for that option in the global options file. If you do not specify options either in the *optionsfile_accessmethod*.opts or in the global option file, the product issues an error message.

For example, if the SAP R/3 access methods are installed for AGENT1 workstation, but you have two external SAP systems on which to schedule jobs, in the <TWA_DIR>/TWS/methods directory, you create the following options files:

- SAP1_AGENT1_r3batch.opts
- SAP2_AGENT1_r3batch.opts

If you have a dynamic pool containing *n* agents, create an options file for the dynamic pool and copy it into the <TWA_DIR>/TWS/methods of each agent of the *n* agents of the pool. For example, if the SAP R/3 access method is installed for **AGENT1** and **AGENT2** which belong to the dynamic pool **DYN_POOL**, create in the <TWA_DIR>/TWS/methods directory of each agent the following options files:

AGENT1

- FILEOPTS_DYN_POOL_r3batch.opts

AGENT2

- FILEOPTS_DYN_POOL_r3batch.opts

The options files must be located in the <TWA_DIR>/TWS/methods directory. They are read when the supported agent is started. Options are specific to each access method. For details about how to configure each access method, see the following sections:

Oracle E-Business Suite
“Configuring the Oracle E-Business Suite access method” on page 52.

PeopleSoft
“Configuring the PeopleSoft access method” on page 61.

SAP R/3
“Configuring the SAP R/3 access method” on page 90.

z/OS “Configuring the z/OS access method” on page 220.

To modify a local or global options file, or to create a local options file, you can use either a text editor or the Option Editor described in “Using the Option Editor” on page 29.

Before completing the installation, Tivoli Workload Scheduler for Applications optionally starts the Option Editor to let you redefine local options, while some of the values you entered during the installation process are used to create a global options file.

Option value inheritance

This property is currently available for r3batch only. Local options files can inherit existing values from the same options in the global options file r3batch.opts. For an access method, the options are listed twice; once as global options and once as local options. If the local options file does not contain a value for the option, then the value for that option in the global options file is used. Otherwise the option value in the local options file is always used.

For example, you might want to define the same value for the **Ljuser** option and a different value for the **retrieve_joblog** option. To do this, you define the **Ljuser** option value in the `r3batch.opts` file. Then you define a different value for the **retrieve_joblog** option in each local options file. This results in the following actions when launching the SAP R/3 job:

- The value for the **Ljuser** option is extracted from the r3batch.opts file.
- The value for the **retrieve_joblog** option is taken from each local options file.

Using the Option Editor

Using the Option Editor.

How you use the Option Editor to perform the following tasks:

- Modify local or global options file, see “Modifying local or global options files” on page 31.
- Create a local options file, see “Creating local options files” on page 31.

To start the Option Editor, go to the `TWS_home/methods/opted` directory of your Tivoli Workload Scheduler installation and, depending on your operating system, run the following command:

On Windows operating systems

```
opted.bat
```

On UNIX operating systems

```
opted.sh
```

Note: To use the Option Editor on a UNIX workstation, you must have a graphical environment.

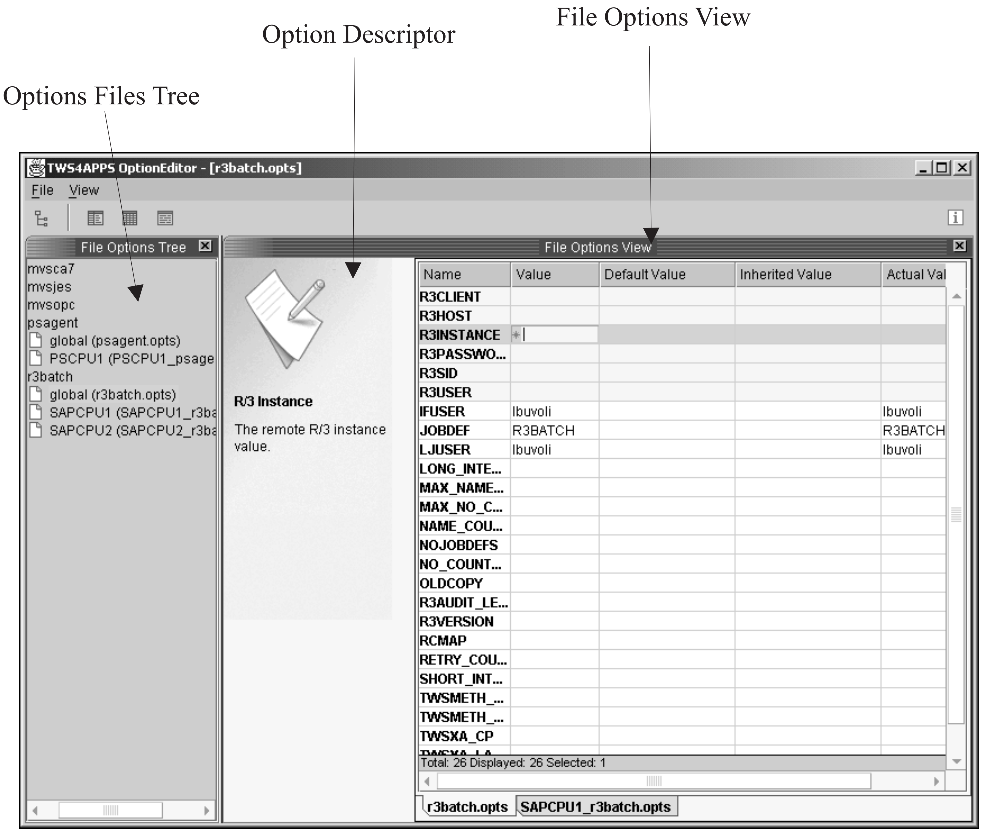
The Option Editor automatically loads all the existing global and local options files grouped by access method.

The Option Editor window has three main areas, as shown in Figure 2 on page 30. To choose which areas you want to see, select **View** > **Show**. The areas are:

File Options Tree
Shows a list of all the existing options files, grouped by method.

Option Descriptor
Gives a description of the current selection.

File Options View
Displays the options of a selected file in any of the three available views.



mvsca7
mvsjes
mvsopc
psagent
global (psagent.opts)
PSCPU1 (PSCPU1_psage
r3batch
global (r3batch.opts)
SAPCPU1 (SAPCPU1_r3ba
SAPCPU2 (SAPCPU2_r3ba

R/3 Instance

The remote R/3 instance value.

Name	Value	Default Value	Inherited Value	Actual Val
R3CLIENT				
R3HOST				
R3INSTANCE				
R3PASSWO...				
R3SID				
R3USER				
IFUSER	lbuvoli			lbuvoli
JOBDEF	R3BATCH			R3BATCH
LJUSER	lbuvoli			lbuvoli
LONG_INTE...				
MAX_NAME...				
MAX_NO_C...				
NAME_COU...				
NOJOBDEFS				
NO_COUNT...				
OLDCOPY				
R3AUDIT_LE...				
R3VERSION				
RCMAP				
RETRY_COU...				
SHORT_INT...				
TWSMETH_...				
TWSMETH_...				
TWSXA_CP				

Total: 26 Displayed: 26 Selected: 1

Figure 2. The Option Editor

The Option Editor provides three possible views of an options file. To change the view in which the options file is displayed select **View** > **Display As**. The views are:

Simple
Displays the options as a sequence of tabs that you can select to edit one option at a time. To view or edit an option, select the tab with the option name to display the field for the value. This is the easiest way to edit options, because you only see the actual value that is used in the file. Inherited or default values are not displayed.

Table Provides a tabular view of all the options for a selected file. This is the default view. For each option the following columns are displayed:

Name The name of the option.

Value The value specified in the file.

Default Value
The value used by the method if not specified in the options file or inherited.

Inherited Value
The value obtained from the global options file if inheritance is supported by the access method. For a detailed explanation, see "Option value inheritance" on page 28.

Actual Value
The value used at run time. The values are used in the order: value, if available; inherited value, if supported; default value.

Text Displays an options file in the typical format, showing only the options that have a value. This view is generally used for preview purposes, because files are shown in read-only form. Password fields are shown encrypted.

Mandatory options are identified by a yellow background and are marked with an asterisk (*). Options that are not correctly entered are shown with a red background. The Option Editor performs only syntactic checks.

Ensure that the Option Editor is not running before you add or update an existing Tivoli Workload Scheduler for Applications installation.

Restrict access to the Option Editor. If you have multiple installations of a supported agent, you can increase security by maintaining only one copy of the Option Editor, for example, on the master workstation.

Modifying local or global options files:
To modify local or global options files, using the Option Editor, perform the following steps:

1. From the **File Options Tree** area, select the options file that you want to modify. The options are displayed in the **File Options View** area.
2. Select the option that you want to modify and modify its value.
3. Save and close the options file.

Creating local options files:
While global options files are created each time you install an access method, you create local options files when you define a new supported agent workstation. To create a local options file, using the Option Editor, perform the following steps:

1. Click **File** > **New** in the menu bar. The New Option File window is displayed.
2. In the **Insert XA CPU Name** field:

   Extended agents
   Enter the name of the extended agent workstation *XANAME* for which you want to create an options file.

   Dynamic agents and Tivoli Workload Scheduler for z/OS agents

   optionsfile
   The name of the options file. This value can be different from the dynamic agent and Tivoli Workload Scheduler for z/OS agent workstation name.
3. Select an access method from the **Insert Method Name** drop-down menu.

   Extended agents

   accessmethod
   The access method you selected.

   Dynamic agents and Tivoli Workload Scheduler for z/OS agents

   accessmethod
   The access method you selected.
4. Enter values for the options you want to define.
5. Save and close the options file.

Defining supported agent workstations

A workstation definition is required for each entity of an access method through which Tivoli Workload Scheduler schedules and launches jobs. For further details about supported agents, see “Supported agent workstations” on page 2.

Creating a workstation using the Dynamic Workload Console

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

Extended agents

To define an extended agent workstation using the Dynamic Workload Console, perform the following steps:

1. From the Dynamic Workload Console portfolio, click **Scheduling environment** > **Design** > **Create Workstations**.
2. Select an engine, distributed or z/OS, from the list and click **Create Workstation**.
3. In the Workstations properties panel, specify the attributes for the extended agent workstation you are creating. For all the details about available fields and options, see the online help by clicking the "?" in the top-right corner. In the workstation definition, specify the access method and other properties, as shown in Table 7. For further information about the workstation definition properties, see *Tivoli Workload Scheduler User’s Guide and Reference*
4. To assign the workstation to an existing domain or to create a new domain, click **Assign to Domain**.
5. Click **Save**.

The following table shows how to complete some specific fields of the workstation properties panel for **extended agents**.

Table 7. How to complete the extended agents definition

Field	Description by Access Method			
	PeopleSoft	**z/OS**	**Oracle E-Business Suite**	**SAP R/3**
Name	The name for the extended agent workstation. For all access methods (except for z/OS, which always has a limit of 8 characters), the name must start with a letter and can contain alphanumeric characters, dashes, and underscores. The maximum length is 16 characters. Workstation names must be unique and cannot be the same as workstation class and domain names. Double-byte character set (DBCS), single-byte character set (SBCS), and bidirectional text are not supported. If a workstation name contains these characters and, as a result, the options file name contains the same name, the workstation cannot be validated by the SAP system. For all the access methods, this name must be consistent with the name of the options file associated with the extended agent workstation. That is, if the options file name is `MYXAGENT_`*`accessmethod`*`.opts`, then `MYXAGENT` and **Name** must be the same.			
Node Name	**null**	The node name or IP address of the z/OS system. Fully qualified domain names are accepted.	**null**	**null**

Table 7. How to complete the extended agents definition (continued)

Field	Description by Access Method			
	PeopleSoft	**z/OS**	**Oracle E-Business Suite**	**SAP R/3**
TCP Port	Any number other than 0.	The TCP/IP address (port number) of the z/OS gateway on the z/OS system. Enter the same value as the SYSTSIN variable PORT.	Any number other than 0.	Any number other than 0.
Access Method	**psagent**	Depending on your job scheduling interface, one of the following: **mvsjes** To launch and monitor z/OS jobs using JES2 or JES3. **mvsopc** To launch and monitor z/OS jobs using Tivoli Workload Scheduler for z/OS. **Note:** In UNIX operating systems, the name is case sensitive and must be lowercase.	**MCMAGENT** **Note:** The name is case sensitive and must be uppercase.	**r3batch** **Note:** In UNIX the name is case sensitive and must be lowercase.

Dynamic agents and Tivoli Workload Scheduler for z/OS agents

The agents are automatically registered to the Tivoli Workload Scheduler network. For further information about the dynamic agents registration, see *Tivoli Workload Scheduler User's Guide and Reference*.

Creating a workstation using the command line

You can define supported agents workstations also using the **composer** command line of Tivoli Workload Scheduler.

Extended agents

The following example shows a definition for a z/OS extended agent workstation named *MVSCPU* that uses the mvsjes access method.

```
cpuname MVSCPU description "zOS extended agent"
     os other
     node mvsesa36.rome.tivoli.com
     tcpaddr 5000
     domain masterdm
     for maestro
        type x-agent
        host ROCIOUS
        access mvsjes
end
```

Dynamic agents

The following example shows a definition for a dynamic agent workstation named *LINUX248* that uses the secure protocol *https* to connect to the Broker server.

```
CPUNAME LINUX248
  DESCRIPTION "This workstation was automatically created."
  OS UNIX
  NODE linux248.romelab.it.ibm.com SECUREADDR 31114
  TIMEZONE Europe/Rome
  FOR MAESTRO HOST NC118003_DWB
    AGENTID "FD640FCA740311E18C4EE96D727FA991"
    TYPE AGENT
    PROTOCOL HTTPS
END
```

For details about defining workstations with **composer**, see the *Tivoli Workload Scheduler User's Guide and Reference*.

Defining workstations for end-to-end scheduling

How to create a workstation definition for end-to-end environment.

Scheduling in an end-to-end environment means that in Tivoli Workload Scheduler for z/OS you are scheduling and monitoring jobs that are physically running on Tivoli Workload Scheduler workstations. For the agents supported in the z/OS environment, see "Supported agent workstations" on page 2.

Extended agents

Extended agent workstations must be defined as fault-tolerant workstations in Tivoli Workload Scheduler for z/OS.

A fault-tolerant workstation is the Tivoli Workload Scheduler for z/OS definition of an existing Tivoli Workload Scheduler agent in the distributed network. The Tivoli Workload Scheduler agent is where the job associated with the fault-tolerant workstation actually runs in the distributed network.

To define the extended agent workstation in Tivoli Workload Scheduler for z/OS, you must:

1. Define the workstation in the CPUREC initialization statement. For an example, see "Creating the CPUREC statement for extended agents."
2. Add the same workstation definition to the database using ISPF or the Dynamic Workload Console. For a description of how to define the workstation using the Dynamic Workload Console, see *Dynamic Workload Console User's Guide*. For an example, see "Defining the workstation with ISPF" on page 35.

Tivoli Workload Scheduler for z/OS agents

To define the agent workstation with z-centric capability in Tivoli Workload Scheduler for z/OS:

1. Add the workstation definition to the database using ISPF or the Dynamic Workload Console. For further information, see *IBM Tivoli Workload Scheduler for z/OS Scheduling End-to-end with z-centric capabilities*.

Creating the CPUREC statement for extended agents:

This section is valid only for Extended agents. Create the CPUREC statement for the workstation in the TOPOLOGY initialization statement. The TOPOLOGY initialization statement is used to define parameters related to the topology of the connected Tivoli Workload Scheduler network. Such a network topology statement is made up of one or more (one for each domain) DOMREC statements that describe the topology of the distributed network, and by several CPUREC statements, one for each fault-tolerant workstation.

The following example shows a CPUREC statement for an SAP R/3 extended agent workstation named R3XA. The extended agent is hosted by a Tivoli Workload Scheduler agent named TWSA, which is also the domain manager of DOMAIN1.

```
****************TPLGINFO MEMBER ***************************

/*********************************************************************/
/* DOMREC: Domain definition                                         */
/*********************************************************************/
DOMREC   DOMAIN(DOMAIN1)
         DOMMNGR(TWSA)
         DOMPARENT(MASTERDM)
/*********************************************************************/
/* CPUREC: Extended agent workstation definition                     */
/*********************************************************************/
CPUREC   CPUNAME(R3XA)
         CPUOS(OTHER)
         CPUNODE(NODE1)
         CPUDOMAIN(DOMAIN1)
         CPUHOST(TWSA)
         CPUTYPE(XAGENT)
         CPUACCESS(r3batch)
         CPUUSER(TWSuser)
         CPUTZ('Europe/Rome')
/*********************************************************************/
/* CPUREC: Domain manager workstation definition                     */
/*********************************************************************/
CPUREC   CPUNAME(TWSA)
         CPUNODE(NODE1)
         CPUAUTOLINK(ON)
         CPUDOMAIN(DOMAIN1)
         CPUTYPE(FTA)
         CPUUSER(TWSuser)
         CPUTZ('Europe/Rome')
```

The following keywords define R3XA as an extended agent:

CPUACCESS
The extended agent access method. For SAP R/3, it is r3batch.

CPUHOST
The name of the Tivoli Workload Scheduler workstation hosting the extended agent. It cannot be another standard agent or extended agent.

CPUTYPE
The workstation type. For an extended agent, it must be XAGENT.

Note: The CPUREC statement does not exist for a Tivoli Workload Scheduler for z/OS agent workstation.

For further information about CPUREC for extended agents, see *Tivoli Workload Scheduler for z/OS Customization and Tuning*.

Defining the workstation with ISPF:

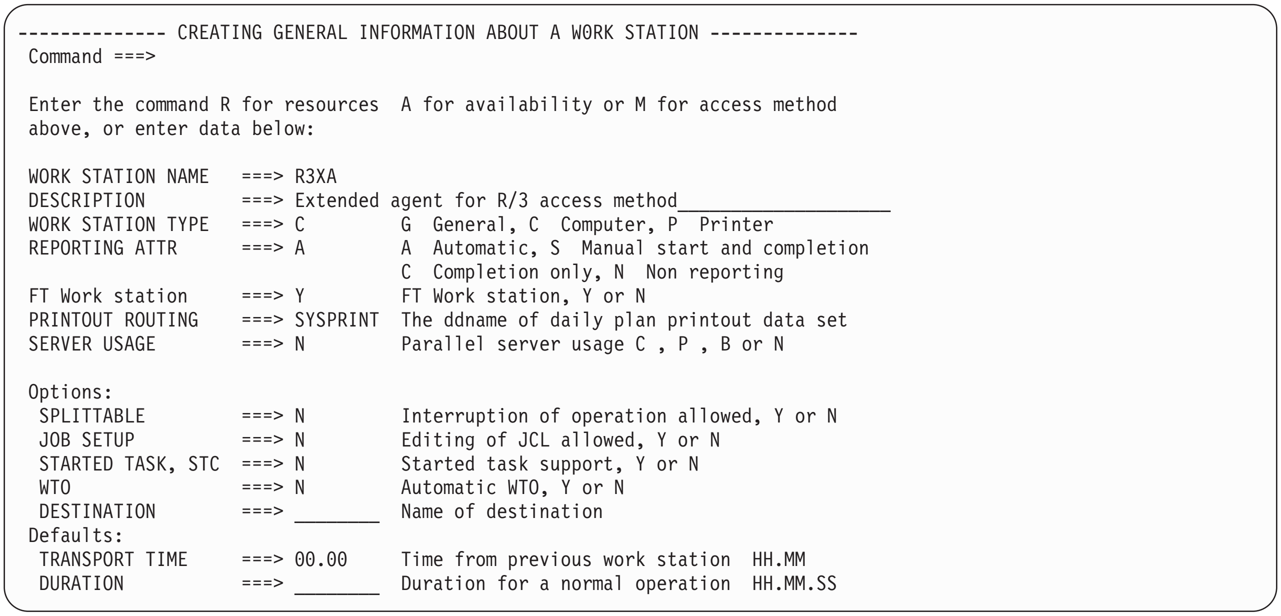
How to create a workstation definition with ISPF.

This section shows the ISPF definition for extended agents and agents with z-centric capability.

Extended agents

In ISPF, define the workstation as computer automatic and then set the *FT Work station* field to **Y**. The CPUREC statement with the three keywords described in “Creating the CPUREC statement for extended agents” on page 34 provides the extended agent specification.

Note: Make sure you write the CPUREC statement before making the ISPF or Dynamic Workload Console definition, because they have no effect without the CPUREC statement.



```
-------------- CREATING GENERAL INFORMATION ABOUT A WORK STATION --------------
 Command ===>

 Enter the command R for resources  A for availability or M for access method
 above, or enter data below:

 WORK STATION NAME   ===> R3XA
 DESCRIPTION         ===> Extended agent for R/3 access method____________________
 WORK STATION TYPE   ===> C         G  General, C  Computer, P  Printer
 REPORTING ATTR      ===> A         A  Automatic, S  Manual start and completion
                                    C  Completion only, N  Non reporting
 FT Work station     ===> Y         FT Work station, Y or N
 PRINTOUT ROUTING    ===> SYSPRINT  The ddname of daily plan printout data set
 SERVER USAGE        ===> N         Parallel server usage C , P , B or N

 Options:
  SPLITTABLE         ===> N         Interruption of operation allowed, Y or N
  JOB SETUP          ===> N         Editing of JCL allowed, Y or N
  STARTED TASK, STC  ===> N         Started task support, Y or N
  WTO                ===> N         Automatic WTO, Y or N
  DESTINATION        ===> ________  Name of destination
 Defaults:
  TRANSPORT TIME     ===> 00.00     Time from previous work station  HH.MM
  DURATION           ===> ________  Duration for a normal operation  HH.MM.SS
```

Figure 3. Defining an Extended Agent workstation

Tivoli Workload Scheduler for z/OS agents

For detailed information and examples about the ISPF definition of Tivoli Workload Scheduler for z/OS agents with z-centric capabilities, see *Scheduling End-to-end with z-centric capabilities*.

Defining jobs for supported agents

To run and monitor a PeopleSoft, SAP R/3, Oracle, or z/OS job with Tivoli Workload Scheduler, the supported agents, or access method require a Tivoli Workload Scheduler job definition, where you specify the external job you want to schedule, the workstation (also defined in Tivoli Workload Scheduler) on which it is to run, and any recovery actions. To define the job, use either of the following methods:

- Dynamic Workload Console.
- Tivoli Workload Scheduler **composer** command line.

If you are scheduling in an end-to-end environment, to define the job, use either of the following methods:

- Dynamic Workload Console.
- Tivoli Workload Scheduler for z/OS ISPF dialogs. You must also create a member in the SCRIPTLIB with a JOBREC statement for the job.

Jobs defined for supported agents are added to job streams and scheduled in the same way as any other job in Tivoli Workload Scheduler and Tivoli Workload Scheduler for z/OS.

Defining jobs with the Dynamic Workload Console

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

Steps for defining a job for supported agents.

Extended agents

To define jobs, follow these steps:

1. From the Dynamic Workload Console portfolio, click **Workload** > **Design** > **Create Workload Definitions**.
2. Specify an engine name, either distributed or z/OS. The Workload Designer opens. Job types and characteristics vary depending on whether you select a distributed or a z/OS engine.
3. In the Working List pane, select **New** > **Job Definition**.
4. Select the category and type of job you want to create.
   - For SAP jobs, **ERP** > **SAP Job on XA Workstations**. See "Creating SAP Standard R/3 jobs from the Dynamic Workload Console" on page 106.
   - For Oracle E-Business Suite, and PeopleSoft, **Native** > **Other**.
   - For z/OS, **Native** > **z/OS**.
5. In the properties panel, specify the attributes for the job definition you are creating. For all the details about available fields and options, see the online help by clicking the "?" in the top-right corner.
6. Click **Save** to save the job definition in the database.

Dynamic agents and Tivoli Workload Scheduler for z/OS agents

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

For all access methods, follow these steps:

1. From the Dynamic Workload Console portfolio, click **Workload** > **Design** > **Create Workload Definitions**.
2. Specify an engine name, either distributed or z/OS. The Workload Designer opens. Job types and characteristics vary depending on whether you select a distributed or a z/OS engine.
3. In the Working List pane, select

   In a distributed environment

   New > **Job Definition**

   In a z/OS environment

   New

   .
4. Select the category and type of job you want to create.
   - For SAP R/3 jobs,

     In a distributed environment

     ERP > **SAP Job on Dynamic Workstations**. See "Creating SAP Standard R/3 jobs from the Dynamic Workload Console" on page 106.

     In a z/OS environment

     ERP > **SAP**. See "Creating SAP Standard R/3 jobs from the Dynamic Workload Console" on page 106.

- For Oracle E-Business Suite, and PeopleSoft, **ERP** > **Access Method**.
- For z/OS, **Native** > **z/OS**.
- For InfoSphere DataStage, see "Defining a Tivoli Workload Scheduler job to run an InfoSphere DataStage job" on page 238.
- For Cognos,see "Defining jobs by using composer" on page 254.
- For SAP PI Channel, see "Defining a Tivoli Workload Scheduler job that runs an SAP PI Channel job" on page 269.

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

Note: The access method for SAP R/3 provides supplementary features if you use the alternative steps described in "Create an SAP job and associate it to a Tivoli Workload Scheduler job" on page 107 or "Creating a SAP job from the Dynamic Workload Console" on page 110. You can create native SAP R/3 Standard jobs on a remote SAP system directly from the Dynamic Workload Console.

Defining jobs using the command line

You can also define jobs using the **composer** command line of Tivoli Workload Scheduler.

Extended agents

The following example describes a Tivoli Workload Scheduler job named `orajob2` that runs on an Oracle Applications extended agent workstation named `ora002`. Tivoli Workload Scheduler logs on to UNIX operating system as `orajobs` and launches a job under Oracle Applications. The Oracle Applications job is named `poxacr` and its owner is `global`. If recovery is needed, Tivoli Workload Scheduler runs job `recov2` and then continues processing.

```
ora002#orajob2
  streamlogon orajobs
  scriptname    "-user global
   -job fnd 'application developer' po poxacr
                  -prn ps4 2
                  -v1 'abc'"
  description "oracle apps job #2"
  recovery continue after recov2
```

The arguments of `scriptname` differ by application. For details, see:

- "Task string parameters for Oracle E-Business Suite jobs" on page 56.
- "Task string parameters for PeopleSoft jobs" on page 67.
- "Task string to define SAP jobs" on page 115.
- "Task definition syntax for z/OS jobs scheduled with Tivoli Workload Scheduler" on page 222.

Dynamic agents

The following example describes a Tivoli Workload Scheduler job named `DYN_JOB_R3_0001` that runs on a dynamic agent workstation named `NC112015_1`. The Tivoli Workload Scheduler launches a job in a SAP environment named `JOB_APPS_86`.

```
NC112015_1#DYN_JOB_R3_0001
 TASK
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition
     xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
     xmlns:jsdlxa="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdlxa"
                  name="r3">
  <jsdl:application name="r3" plugin="xajob">
    <jsdlxa:xajob accessMethod="r3batch" target="NW73LIN_r3batch">
            <jsdlxa:taskString>/ -job JOB_APPS_86 -i 14514200 -c c
                 -flag ENABLE_APPL_RC </jsdlxa:taskString>
        </jsdlxa:xajob>
  </jsdl:application>
</jsdl:jobDefinition>
 RECOVERY STOP
```

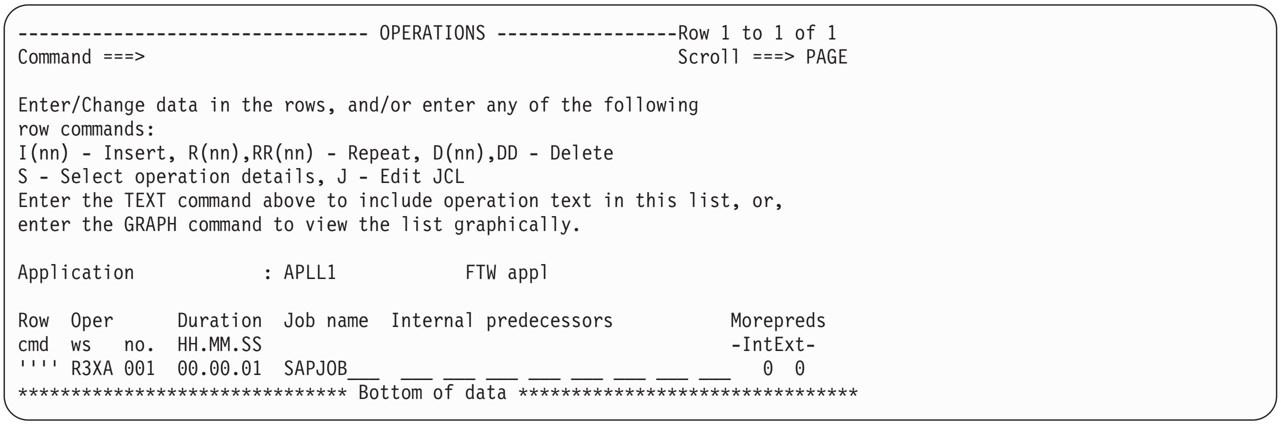
For more information about using the **composer** command line to define jobs, see *Tivoli Workload Scheduler: User's Guide and Reference*.

Defining jobs for end-to-end scheduling

Extended agents

Extended agent jobs scheduled to run in an end-to-end environment cannot be defined using the Dynamic Workload Console or the Tivoli Workload Scheduler command line, but must be added to the SCRIPTLIB of Tivoli Workload Scheduler for z/OS.

In the OPERATIONS ISPF panel of Tivoli Workload Scheduler for z/OS, extended agent jobs are defined like any other job, but with the specific attribute for a job defined on an extended agent workstation. The following example shows the definition of a job named SAPJOB. This is the Tivoli Workload Scheduler for z/OS job that drives the running of on SAP R/3 job (named BAPRINT46B as shown in the next example). It shows as an extended agent job because the associated workstation is an extended agent workstation named R3XA.



```
--------------------------------- OPERATIONS -----------------Row 1 to 1 of 1
Command ===>                                                  Scroll ===> PAGE

Enter/Change data in the rows, and/or enter any of the following
row commands:
I(nn) - Insert, R(nn),RR(nn) - Repeat, D(nn),DD - Delete
S - Select operation details, J - Edit JCL
Enter the TEXT command above to include operation text in this list, or,
enter the GRAPH command to view the list graphically.

Application            : APLL1           FTW appl

Row  Oper      Duration  Job name  Internal predecessors           Morepreds
cmd  ws   no.  HH.MM.SS                                            -IntExt-
'''' R3XA 001  00.00.01  SAPJOB___  ___ ___ ___ ___ ___ ___ ___ ___   0  0
******************************* Bottom of data ********************************
```

Figure 4. Defining an Extended Agent job for end-to-end scheduling

For each job, create a member in the SCRIPTLIB of Tivoli Workload Scheduler for z/OS with details about the job in a JOBREC statement. A SAPJOB member was created for the job of the previous example. It contains a JOBREC statement like this:

```
JOBREC
  JOBCMD('/-job BAPRINT46B -user MAESTRO -i 14160001 -c C')
  JOBUSR(twsila)
```

The string in JOBCMD is read and interpreted by the access method before running the job. The job of this example, BAPRINT46B, was previously defined on SAP R/3 and assigned with an ID of 14160001, that was manually written in JOBCMD.

The following example is for a PeopleSoft job. The entire string that follows the JOBCMD keyword must be enclosed within quotation marks ("), because for PeopleSoft jobs single quotes are already used in the string.

```
JOBREC
   JOBCMD("/ -process XRFWIN -type 'SQR Report' -runcontrol TWS")
   JOBUSR(PsBuild)
```

Tivoli Workload Scheduler for z/OS agents
For information about the jobs definition for agent with z-centric capabilities, see *Scheduling End-to-end with z-centric capabilities*.

The arguments of JOBCMD differ by application. For details, see:

- "Task string parameters for Oracle E-Business Suite jobs" on page 56.
- "Task string parameters for PeopleSoft jobs" on page 67.
- "Task string to define SAP jobs" on page 115 or "Defining SAP jobs dynamically" on page 131.
- "Task definition syntax for z/OS jobs scheduled with Tivoli Workload Scheduler" on page 222.

Submitting jobs

To submit jobs on the supported agent workstation, perform the following steps:

1. Verify that the application system to which the job belongs and the related database is up and running.
2. Launch the job. For details, see:

   Extended agents

   - *Tivoli Workload Scheduler User's Guide and Reference* for **conman** command line.
   - *Dynamic Workload Console User's Guide* for Dynamic Workload Console.

   Dynamic agents

   - *Tivoli Workload Scheduler User's Guide and Reference* for **conman** command line.
   - *Dynamic Workload Console User's Guide* for Dynamic Workload Console.

   Tivoli Workload scheduler for z/Os agents

   - *IBM Tivoli Workload Scheduler for z/OS Scheduling End-to-end with z-centric capabilities* for **ISPF** panel.
   - *Dynamic Workload Console User's Guide* for Dynamic Workload Console.

Configuring FIPS mode

Configure IBM Tivoli Workload Scheduler for Applications to use Federal Information Processing Standards (FIPS) cryptographic modules.

Federal Information Processing Standards (FIPS) are standards and guidelines issued by the National Institute of Standards and Technology (NIST) for federal computer systems. FIPS are developed when there are compelling federal government requirements for standards, such as for security and interoperability, but acceptable industry standards or solutions do not exist. IBM Tivoli Workload Scheduler for Applications can be configured to use the FIPS 140-2 approved cryptographic provider(s); IBMJCEFIPS (certificate 376) and/or IBMJSSEFIPS (certificate 409) and/or IBM Crypto for C (ICC (certificate 384) for cryptography. The certificates are listed on the NIST web site at http://csrc.nist.gov/cryptval/140-1/1401val2004.htm. For more information about FIPS crytographic modules, refer to the *Administration Guide*.

To configure IBM Tivoli Workload Scheduler for Applications to use FIPS cryptographic modules, perform the following steps:

Edit the SDK master security properties file, `java.security`, located in the `TWS_HOME/methods/_tools/_jvm/jre/lib/security` directory to insert the IBMJCEFIPS provider (com.ibm.crypto.fips.provider.IBMJCEFIPS). IBMJCEFIPS must precede the IBMJCE provider in the provider list.
The following is an example of the edited SDK java.security file:

```
security.provider.1=com.ibm.crypto.fips.provider.IBMJCEFIPS
security.provider.2=com.ibm.crypto.provider.IBMJCE
security.provider.3=com.ibm.jsse.IBMJSSEProvider
security.provider.4=com.ibm.jsse2.IBMJSSEProvider2
security.provider.5=com.ibm.security.jgss.IBMJGSSProvider
security.provider.6=com.ibm.security.cert.IBMCertPath
security.provider.7=com.ibm.crypto.pkcs11.provider.IBMPKCS11
security.provider.8=com.ibm.security.cmskeystore.CMSProvider
security.provider.9=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
```

The following is an example of the edited java.security file if you are using the Java Virtual Machine for HP-UX or Solaris:

```
security.provider.1=sun.security.provider.Sun
security.provider.2=com.ibm.crypto.fips.provider.IBMJCEFIPS
security.provider.3=com.ibm.crypto.provider.IBMJCE
security.provider.4=com.ibm.jsse.IBMJSSEProvider
security.provider.5=com.ibm.jsse2.IBMJSSEProvider2
security.provider.6=com.ibm.security.jgss.IBMJGSSProvider
security.provider.7=com.ibm.security.cert.IBMCertPath
security.provider.8=com.ibm.i5os.jsse.JSSEProvider
#security.provider.8=com.ibm.crypto.pkcs11.provider.IBMPKCS11
security.provider.9=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
```

To begin working in FIPS operational mode, enable the **`use_fips`** option in the common options file. See “Defining the common options” on page 94 and Table 13 on page 53.

Installing and uninstalling IBM Tivoli Workload Scheduler for Applications Plug-ins

This chapter describes how to install and uninstall IBM Tivoli Workload Scheduler for Applications Plug-ins.

Planning the installation

Refer to the following sections to plan your installation of Tivoli Workload Scheduler for Applications plug-ins.

Supported operating systems

For a complete list of supported operating systems, perform the following actions to produce a dynamic report:

1. Go to the web page: http://publib.boulder.ibm.com/infocenter/prodguid/v1r0/clarity/index.html.
2. Click the *"Operating systems for a specific product"* report.

3. In the windows "Operating systems for a specific product", fill in the field *Enter a full or partial product name* with the value **IBM Tivoli Workload Scheduler for Applications** and click *Search*.
4. In the *Select version* drop-down list, select the version **8.6** and click *Submit to run the report*.

For a complete list of systems requirements (disk space, temporary space and RAM usage), see http://www.ibm.com/support/docview.wss?rs=672&uid=swg27023736.

Software prerequisites

IBM Tivoli Workload Scheduler for Applications Plug-ins version 8.6 prerequisites Tivoli Workload Scheduler version 8.6 Fix Pack 1 , or later.

IBM Tivoli Workload Scheduler for Applications Plug-ins version 8.6 interacts with the following products:

- InfoSphere DataStage. For more information, see "Software requirements" on page 235.
- Cognos. For more information, see "Software requirements" on page 249.
- SAP PI Channel. For more information, see "Software requirements" on page 268.

Preparing your environment

Before you install the product, ensure that:

- You have a working installation of Tivoli Workload Scheduler version 8.5.1, or later, installed on your workstation.
- You have root permissions on UNIX and administrator permissions on Windows to run the installation.
- There are no scheduling and monitoring processes currently running on the FTA where you want to perform the installation.
- For the access method for PeopleSoft, the corresponding PeopleSoft environment is already installed and fully accessible from the workstation where you are installing.
- For the access method for Oracle E-Business Suite, the Oracle E-Business Suite server is already installed on the same physical node where the access method and the Tivoli Workload Scheduler host are located.
- The installation process is not already running on the workstation. You can verify this by checking if the setup process is running.
- The Option Editor is not running.
- The X Server is installed on a workstation in your environment.
- The X-windows Display is appropriately exported if you are launching the installation process using a telnet session.
- If you are installing the product, you have the disk space and the temporary space indicated in http://www.ibm.com/support/docview.wss?rs=672&uid=swg27023736.

The installation DVD

The following list shows the installation DVDs contained in Tivoli Workload Scheduler for Applications divided by platform:

- IBM Tivoli Workload Scheduler for Applications 8.6 for AIX
- IBM Tivoli Workload Scheduler for Applications 8.6 for HP-UX for Integrity
- IBM Tivoli Workload Scheduler for Applications 8.6 for HP-UX for PA-RISC
- IBM Tivoli Workload Scheduler for Applications 8.6 for Linux

- IBM Tivoli Workload Scheduler for Applications 8.6 for Linux S390
- IBM Tivoli Workload Scheduler for Applications 8.6 for Sun Solaris
- IBM Tivoli Workload Scheduler for Applications 8.6 for Sun Solaris for Opteron
- IBM Tivoli Workload Scheduler for Applications 8.6 for Windows

Each DVD includes the following directories:
- *operating system* specific files
- INSTALLER
- LICENSE
- RESPONSE_FILE
- SAMPLES
- LAUNCHPAD
- z/OS libraries for the specific operating systems
- CLI
- GSKIT
- JVM_64

  Note: On Sun Solaris for Opteron and HP-UX for Integrity, this folder is not included.
- PLUGINS
- PLUGIN_INSTALLER

Installing

The plug-ins you install with IBM Tivoli Workload Scheduler for Applications Plug-ins.

Depending on IBM Tivoli Workload Scheduler for Applications Plug-ins, you want to install, you can interact with the following products:
- InfoSphere DataStage. For more information, see “Software requirements” on page 235
- Cognos. For more information, see “Software requirements” on page 249.
- SAP PI Channel. For more information, see “Software requirements” on page 268.

Note: The installation ofIBM Tivoli Workload Scheduler for Applications Plug-ins version 8.6.0. does not update the Software Distribution catalogs.

Throughout this guide, the path where you install IBM Tivoli Workload Scheduler for Applications Plug-ins is called `TWA_home`. The default installation path depends on the Tivoli Workload Scheduler component where you install IBM Tivoli Workload Scheduler for Applications Plug-ins, for example the Master Domain Manager or an agent. For detailed information about the default installation path of the Tivoli Workload Scheduler components, see the *IBM Tivoli Workload Scheduler: Planning and Installation* manual related to the product version you are using.

Before installing

Before you install IBM Tivoli Workload Scheduler for Applications Plug-ins, ensure that the following conditions are satisfied:
- To extract the eImages released in **zip** format, see “Before installing” on page 6.
- You have the following permissions to run the installation:
  Windows operating systems:
  Administrator

UNIX and Linux operating systems:
Root

- The installation process is not already running on the workstation. You can verify it by checking that the **setup** process is not running.

Selecting your installation method

You can install IBM Tivoli Workload Scheduler for Applications Plug-ins by using one of the methods described in this section. To install one or more plug-ins, use any of the following procedures. If you didn't installed all plug-ins and, later on, you want to add another one, ensure that you use the same installation procedure that you used initially.

Installation wizard
Install IBM Tivoli Workload Scheduler for Applications Plug-ins or add a plug-in to an existing installation by running the individual setup files for each supported operating system. For details, see "Installing by using the installation wizard."

Silent installation
Customize a *response file* by adding all the configuration settings to be used during installation. Then, from the command line, run the setup command. With this procedure, you can run the installation unattended and in the background. For details, see "Installing by using the silent installation" on page 45.

Note: To successfully use the installed plug-ins, you must first restart WebSphere Application Server, Tivoli Workload Scheduler agent or both.

Installing by using the installation wizard

To install IBM Tivoli Workload Scheduler for Applications Plug-ins, perform the following steps:

1. From the product DVD for the operating system where you are installing the product, or from an eimage, run the **setup** installation command. It is located in the `*operating_system*/INSTALLER` directory. The installation starts.
2. Select the language in which you want the wizard to be displayed, and click **OK**. The Welcome panel is displayed.
3. Read the welcome information and click **Next**. The Software License Agreement panel is displayed.
4. Read the Software License Agreement information and select the radio button to accept the license agreement. Click **Next**. The Discovery instance panel is displayed with all the Tivoli Workload Automation instances found on your workstation.
5. Select the instance path where you want to install the plug-ins.

   Install IBM Tivoli Workload Scheduler for Applications Plug-ins 8.6
   To install one or more IBM Tivoli Workload Scheduler for Applications Plug-insplug-ins.

   If the installation program does not detect any Tivoli Workload Scheduler installation on the instance path you selected, you cannot perform any actions.
6. Click **Next**. The installation program lists all the plug-ins you can install.
7. Make as many selections as you want, and click **Next**.
8. Enter the requested information and click **Next**. A summary information panel is displayed.

9. Click **Next**. The installation process begins; the progress panel is displayed showing the status.
10. Click **Done** to exit the installation wizard and complete the installation.

If you received error messages, analyze the installation log files shown in Table 8.

Table 8. Installation log files

Log file name	Content	Directory
`tws4apps_status.log`	Tivoli Workload Scheduler for Applications installation status log file. It reports if the installation completed successfully or with errors. In case of errors it indicates if the error is due to an incorrect field value or to a failed step.	*Tivoli Workload Automation _installation_dir*\logs
`tws4apps_ia_install.log`	Tivoli Workload Scheduler for Applications installation log file for InstallAnywhere errors.	*Tivoli Workload Automation_installation_dir*\logs
`tws4apps_install.log`	Tivoli Workload Scheduler for Applications installation log file.	*Tivoli Workload Automation_installation_dir*\logs

Installing by using the silent installation

A silent installation runs according to the parameters set in a response file. The response file includes all the installation information required to run the installation without user intervention.

To install IBM Tivoli Workload Scheduler for Applications Plug-ins with the silent installation, you are provided with the following response file located under `RESPONSE_FILE` on the product DVD:

```
TWSAPPS_RespFile_<operatingsystem>.txt
```

where <*operatingsystem*> can be UNIX or Windows.

It is a template file that you can customize to reflect the type of installation you want to perform.

Using the silent installation you can install one or more plug-ins simultaneously.

When running the installation in silent mode, no messages are displayed. The messages are written in the silent installation log files listed in Table 8. If the silent installation fails, you can verify the messages written in the log files, by checking them in the section"Analyzing return codes for silent installation" on page 283.

To run the silent installation, perform the following steps:

1. Create your response file or customize the response file to include the options required to complete the installation.

   ```
   TWSAPPS_RespFile_<operatingsystem>.txt
   ```

   For a list of these options, see Table 9 on page 46.

   The response file must be accessible from the workstation where you want to perform the installation. Entries in the response file are in the format *option=value*. Each entry must be written on a separate line.
2. Insert the product DVD for your operating system and run the **setup** command, located in the `operating_system/INSTALLER` directory:

   On UNIX and Linux operating systems:

```
./setup.sh -i silent -f response_file
```

On Windows operating systems:

```
setup.bat -i silent -f response_file
```

Where:

-i silent
Specifies that the installation is run unattended, driven by a response file.

-f *response_file*
Indicates the fully qualified path to the response file that contains the installation options. *response_file* can be any text file with the name and extension you choose.

Table 9 lists the options you can specify to drive the installation.

Table 9. Options to perform a silent installation

Option	Required	Description	Value

USER_INSTALL_DIR=<*path*>	Yes	Specify the Tivoli Workload Scheduler installation path where you want to install IBM Tivoli Workload Scheduler for Applications Plug-ins.	A fully qualified path. For example, to install the product under `c:\program Files\IBM\TWA86`, specify:
			`USER_INSTALL_DIR= c:\program Files\IBM\TWA86`
			The product files are installed in
			`c:\program Files\IBM\TWA86\TWS\methods`
			On Windows operating systems: The default path is **`c:\\Program Files\\IBM\\TWA`**
			On UNIX and Linux operating systems: The default path is**`/opt/IBM/TWA`**

Table 9. Options to perform a silent installation (continued)

Option	Required	Description	Value
CHOSEN_INSTALL_FEATURE_LIST=<*value1*>, ...,<*valuen*>	Yes	Specify the values that correspond to the plug-in you want to install, separated by comma. Possible values are:	**CGN** To install Cognos plug-in. **DTS** To install InfoSphere DataStageplug-in. **PIC** To install SAP PI Channel plug-in. For example, to install the Cognos and the InfoSphere DataStage plug-ins, specify the following values: `CHOSEN_INSTALL_FEATURE_LIST=CGN,DTS`

The following is an example of the command you run to perform a silent installation on a UNIX workstation, by using the response file `TWSAPPS_RespFile_UNIX.txt`:

```
./setup.sh -i silent -f /tmp/TWSAPPS_RespFile_UNIX.txt
```

The following example shows a response file that installs the Cognos and the InfoSphere DataStage plug-ins on a Windows workstation:

```
USER_INSTALL_DIR=c:\\Program Files\\IBM\\TWA
CHOSEN_INSTALL_PLUGINS_LIST=CGN,DTS
```

Uninstalling

Uninstalling the product either using the wizard or the silent method.

When you uninstall IBM Tivoli Workload Scheduler for Applications Plug-ins, you can uninstall one or more plug-ins simultaneously.

If you installed IBM Tivoli Workload Scheduler for Applications Plug-ins by using the silent installation, uninstall it using the same procedure.

To uninstall IBM Tivoli Workload Scheduler for Applications Plug-ins, you can use any of the following procedures:

Wizard
For details, see "Uninstalling by using the wizard" on page 48.

Silent For details, see "Uninstalling by using the silent uninstallation" on page 49.

Before uninstalling

Before you uninstall one or more plug-ins, ensure that the following conditions are satisfied:

- You have the following permissions to run the installation:
  Windows operating systems:
  Administrator
  UNIX and Linux operating systems:
  Root

- The installation process is not already running on the workstation. You can verify it by checking that the **setup** processes is not running.
- The plug-ins are not in use by IBM Tivoli Workload Scheduler processes.

Uninstalling by using the wizard

Uninstalling IBM Tivoli Workload Scheduler for Applications Plug-ins by using the wizard.

Using the wizard, you can uninstall one or more plug-ins at a time.

If you installed the product using the installation wizard, you can uninstall it either using the uninstallation wizard or the silent uninstallation. If you installed the product using the silent installation you must use the silent uninstallation to uninstall it.

To uninstall one or more plug-ins, perform the following steps:

1. Run the following command:

   On Windows operating systems:

   From the *TWS_home*\methods directory,

   ```
   .\_tws4apps_uninst\_tws4apps_uninst.bin
   ```

   On UNIX and Linux operating systems:

   From the *TWS_home*/methods directory,

   ```
   ./_tws4apps_uninst/_tws4apps_uninst.bin
   ```

   The uninstall program starts.
2. Select the language for the wizard, and click **OK**. The Welcome window is displayed.
3. Read the welcome information and click **Next**.
4. A summary information window is displayed listing all plug-ins that you can uninstall. By default the product uninstalls all the plug-ins.
5. Select the plug-ins you want to uninstall and click **Next**. The uninstall process starts. When the uninstallation completes, a panel showing the results, is displayed.
6. Click **Finish** to exit the uninstallation wizard.

   Note: If you run the **uninstaller** command from the `_tws4apps_uninst` directory, you must exit from the `_tws4apps_uninst` directory to complete the uninstallation. After the uninstallation completed successfully, remove the `_tws4apps_uninst` directory.

If you received error messages, analyze the uninstallation log files listed in Table 10.

Table 10. Uninstallation log files

Log file name	**Content**	**Directory**
`tws4apps_status.log`	Tivoli Workload Scheduler for Applications installation status log file. It reports if the uninstallation completed successfully or with errors. In case of errors it indicates if the error is due to an incorrect field value or to a failed step.	*Tivoli Workload Automation_installation_dir*\logs
`tws4apps_ia_uninstall.log`	Tivoli Workload Scheduler for Applications log file for InstallAnywhere errors.	*Tivoli Workload Automation_installation_dir*\logs

Table 10. Uninstallation log files (continued)

Log file name	Content	Directory
`tws4apps_uninstall.log`	Tivoli Workload Scheduler for Applications log file.	*Tivoli Workload Automation_installation_dir*\logs

Uninstalling by using the silent uninstallation

Use the silent uninstallation to uninstall the product in silent mode, without user intervention. Using the silent installation you can uninstall all the installed plug-ins simultaneously or one plug-in at a time.

If you installed the product using the installation wizard, you can uninstall it either using the uninstallation wizard or the silent uninstallation. If you installed the product using the silent installation you must use the silent uninstallation to uninstall it.

When running the uninstallation in silent mode, no messages are displayed. The messages are written in the silent installation log files listed in Table 5 on page 23. If the silent uninstallation fails, you can verify the messages written in the log files, by checking them in the section"Analyzing return codes for silent installation" on page 283.

To uninstall one or more plug-ins at a time, run the following command:

On Windows operating systems:

From the *TWS_home*\methods directory,

```
tws4apps_uninst -i silent -f <response file>
```

Where *response file* is a template file that you can customize to indicate the list of the plug-ins you want to uninstall. The default response file is `TWSAPPS_RespFile_Uninst_windows.txt`. It is located under the `RESPONSE_FILE` directory.

On UNIX and Linux operating systems:

From the *TWS_home*/methods directory,

```
./_tws4apps_uninst/_tws4apps_uninst.bin -i silent -f <response file>
```

Where *response file* is a template file that you can customize to indicate the list of the plug-ins you want to uninstall. The default response file is `TWSAPPS_RespFile_Uninst_unix.txt`. It is located under the `RESPONSE_FILE` directory

Table 11 on page 50 lists the options you can specify when uninstalling.

Table 11. Options to perform a silent uninstallation

Option	Required	Description	Value

FEATURE_UNINSTALL_LIST=<*value*>	Yes	Specify the values that correspond to the plug-ins you want to uninstall, separated by comma.	Possible values are:
			Cognos
			To uninstall Cognosplug-in.
			DataStage
			To uninstall InfoSphere DataStageplug-in.
			PIChannel
			To uninstall PIChannel plug-in.
			For example, to uninstall the Cognos and the InfoSphere DataStage plug-ins, specify the following values:
			`PLUGINS_UNINSTALL_LIST=`
			`Cognos,`
			`DataStage`

The following is an example of the command you run from the directory *TWS_home*\methods, to perform a silent installation on a UNIX workstation, by using the response file TWSAPPS_RespFile_UNIX.txt:

```
./_tws4apps_uninst/_tws4apps_uninst.bin -i silent -f TWSAPPS_RespFile_UNIX.txt
```

The following example shows a response file that uninstalls the Cognos and the InfoSphere DataStage plug-ins:

```
FEATURE_UNINSTALL_LIST=Cognos,DataStage
```

Part 2. Oracle E-Business Suite access method

Introducing Tivoli Workload Scheduler for Applications for Oracle E-Business Suite

Using Tivoli Workload Scheduler for Applications for Oracle E-Business Suite you can schedule Oracle E-Business Suite jobs using Tivoli Workload Scheduler.

The Tivoli Workload Scheduler host, Tivoli Workload Scheduler extended agent for Oracle E-Business Suite, and Oracle E-Business Suite instance must be installed on the same physical node.

Software requirements

In addition to the supported operating systems and prerequisite software listed in "Installing, upgrading, and uninstalling IBM Tivoli Workload Scheduler for Applications" on page 3, you must have Oracle Applications version 10.7 or Oracle E-Business Suite versions 11.0 or 11.i installed.

Note: The procedure described in this part refers to Oracle E-Business Suite and applies both to Oracle Applications version 10.7 and to Oracle E-Business Suite versions 11.0 or 11.i, unless otherwise specified.

Features

Using Tivoli Workload Scheduler for Applications for Oracle E-Business Suite, you can perform the following tasks:

- Use Tivoli Workload Scheduler standard job dependencies (except Open File dependencies).
- Schedule Oracle E-Business Suite jobs to run on specified days, times, and in a prescribed order.
- Define inter-dependencies between Oracle Applications jobs and jobs that run on different systems such as SAP R/3 and PeopleSoft.
- Define inter-dependencies between Oracle Applications jobs and jobs that run on different operating systems.
- Customize job execution return codes.

Roles and responsibilities

In a typical enterprise, different users contribute to the implementation and operation of the product. Table 12 describes the roles and responsibilities of all actors in the process model, showing the tasks they perform.

Table 12. Roles and responsibilities in Tivoli Workload Scheduler for Applications for Oracle E-Business Suite

User role	User task
Oracle E-Business Suite administrator	"Completing the installation of the access method for Oracle E-Business Suite" on page 15
Oracle E-Business Suite developer	"Defining jobs in Oracle E-Business Suite" on page 56
Tivoli Workload Scheduler for Applications configurator	"Defining the configuration options" on page 53

Table 12. Roles and responsibilities in Tivoli Workload Scheduler for Applications for Oracle E-Business Suite (continued)

User role	User task
Tivoli Workload Scheduler for Applications developer	"Defining Oracle E-Business Suite jobs in Tivoli Workload Scheduler" on page 56

Interface software

The interface between Tivoli Workload Scheduler and Oracle E-Business Suite comprises the following software:

Front-end

MCMAGENT
The Tivoli Workload Scheduler for Applications for Oracle E-Business Suite access method. Use this to prepare a job for launching. When you launch, monitor and, optionally, stop the job, using the options of the method, the method launches one of the back-end programs to accomplish the task.

Back-end

MCMLJ
A program that submits a job to Oracle Concurrent Manager.

MCMSTAT
A program that monitors the status of the submitted job.

MCMTJ
A program that stops the submitted job on receiving a `kill` signal from Tivoli Workload Scheduler.

mcm.mk
A makefile utility for linking the above programs with Oracle E-Business Suite.

Options

MCMAGENT.opts
A global options file which contains configuration settings for the access method. It is created by the installation process of Tivoli Workload Scheduler for Applications with some of the values you enter during installation. See "Configuring the Oracle E-Business Suite access method" and "Setting options with the Option Editor" on page 26 for details.

*XANAME*_**MCMAGENT.opts**
A local options file which contains configuration settings for the extended agent. One file is required for each extended agent workstation that you define. See "Configuring the Oracle E-Business Suite access method" and "Setting options with the Option Editor" on page 26 for details.

Configuring the Oracle E-Business Suite access method

This chapter provides detailed reference information about the Tivoli Workload Scheduler for Applications for Oracle E-Business Suite options file.

Defining the configuration options

The Tivoli Workload Scheduler for Applications installation process creates a default global options file for the Oracle E-Business Suite access method, named `MCMAGENT.opts`. You can also create local files, named *XANAME*`_MCMAGENT.opts`, which must be located in the *TWS_home*`/methods` directory of the hosting computer for the extended agent.

To edit both options files, you can use either the Option Editor available with this product, or any other text editor. For details about how to create and edit the options files with the Option Editor, see "Setting options with the Option Editor" on page 26. For examples of an options file for this access method, see "Examples" on page 55.

If you do not create a local options file, the agent uses, by default, the global options file.

Table 13 lists the options that you can define for the MCMAGENT access method. The options must be written one per line and have the following format (with no spaces included):

```
OPTION=VALUE
```

Table 13. MCMAGENT access method options

Option	Description
CFUSER	(Mandatory) Assigns the user name used by the access method to check file dependencies (CF tasks). It can be the same as LJUSER.
ENV	(Mandatory) The environment file to run. Each instance of Oracle E-Business Suite has its own environment file (APPLSYS.env or APPDEMO.env). The **`.env`** filename identified here is used to set the environment dynamically when Tivoli Workload Scheduler runs jobs on the Oracle Applications extended agent.
FND_TOPDIR	(Mandatory) The FND top directory where the Oracle E-Business Suite Object Library is located.
MAX_TIME	(Mandatory) This acts as the maximum increment of the job status polling interval. The job status polling interval is incremented each polling cycle (see MIN_TIME). When the interval reaches or exceeds the value of this option, it remains at this value. It is expressed in seconds. Use MIN_TIME and MAX_TIME to poll for status information more frequently when the job has just started (helping you to catch an immediate job failure as soon as possible, for example), and then change to a longer interval for jobs that are expected to take some time before completion.

Table 13. MCMAGENT access method options (continued)

Option	Description
MIN_TIME	(Mandatory) This value is used as both the initial job status polling interval, and as the incremental value by which the polling interval is increased each polling cycle. It is expressed in seconds. Use MIN_TIME and MAX_TIME to poll for status information more frequently when the job has just started (helping you to catch an immediate job failure as soon as possible, for example), and then change to a longer interval for jobs that are expected to take some time before completion.
ORA_APP_TOPDIR	(Mandatory) The UNIX directory where Oracle E-Business Suite is installed.
ORACLE_HOME	(Mandatory) The UNIX directory where Oracle E-Business Suite is installed.
ORACLE_SID	(Mandatory) The Oracle system identifier of the database instance you are connecting to. This corresponds to the value of the Oracle E-Business Suite$ORACLE_SID variable.
PASSWORD	(Mandatory) The password for the selected user ID. It must be encrypted. To encrypt the password use the `enigma` utility, provided with the product. For details, see "Encrypting Oracle E-Business Suite user passwords" on page 55.
PATH	(Mandatory) The setting of the PATH variable, including the existing path, the path to the **`bin`** directory of Oracle E-Business Suite, the path to the **`include`** directory under the FND Top directory, and others as needed.
PRINTER_NAME	(Mandatory) The default printer ID.
THREE_TIER_DB	(Optional) Allows the extended agent to function in a three-tier Oracle connectivity environment. Supply the TNS service name defined in the Oracle environment pointing to a remote database server. See the sample options file shown in "Examples" on page 55.
TWSMETH_CP	(Optional) The code page that the MCMAGENT access method uses to write its output. This option must be consistent with TWSMETH_LANG. It can be any of the existing TIS codepages.
TWSMETH_LANG	(Optional) The language used to report messages. This option must be consistent with TWSMETH_CP.
TWSXA_CP	(Optional) Specifies the encoding character set of the Oracle E-Business Suite database. The default value is the default code set of the operating system.
`use_fips`	(Optional) Enables (ON) or disables (OFF) the FIPS mode of operation for Tivoli Workload Scheduler for Applications. The default value is OFF.
USERID	(Mandatory) A database user ID that has access to all the modules in Oracle E-Business Suite. It must be able to launch, manage, and terminate jobs in Oracle E-Business Suite.

Table 13. MCMAGENT access method options (continued)

Option	Description
UTF8CMDLINE	(Optional) Enables the encoding of the extended parameters in UTF-8 format. Specify *1* to enable the option, or *0* to disable it. The default value is *0*, meaning that the code set of the operating system is used.

Examples

File ORACLECPU_MCMAGENT.opts is a sample options file:

```
ORACLE_HOME=/general/ora716
ORACLE_SID=DEMO106
ORA_APP_TOPDIR=/apps106
FND_TOPDIR=$APPL_TOP/fnd/*
PATH=$PATH:$ORACLE_HOME/bin:$FND_TOP/include
ENV=$APPL_TOP/APPDEMO106.env
USERID=APPS_US
PASSWORD=APPS_US
PRINTER_NAME=PRINTER1
MIN_TIME=10
MAX_TIME=200
```

The following example shows the options file of an extended agent configured for an Oracle three-tier connectivity environment. In this example, `TST` is the name of the Oracle database instance where the Oracle E-Business Suite database is located and `REMOTEDB` is the TNS Oracle service name.

```
MIN_TIME=10
ORACLE_HOME=/oracle/db/TST
ORACLE_SID=TST
ORA_APP_TOPDIR=/oracle/Apps11
FND_TOPDIR=$APPL_TOP/fnd/11.0.28
PATH=$PATH:$FND_TOP/bin:$MCM_TOP/bin
ENV=$APPL_TOP/APPTST.env
USERID=apps
PASSWORD=##3Y-UGcZ-3Kap-6m
PRINTER_NAME=prt4b2
MAX_TIME=150
THREE_TIER_DB=REMOTEDB
```

The following example shows the encoding options to set for an Oracle E-Business Suite database that uses the code page UTF-8 so that it can process a job whose task string contains Japanese characters.

```
UTF8CMDLINE=1
TWSXA_CP=UTF8
```

Encrypting Oracle E-Business Suite user passwords

When you make entries in the options file, the *password* value is automatically encrypted before it is written to the file. If you modify the file with a text editor, you must run the **enigma** program to encrypt the password before writing it to the file. Run the **enigma** program as follows:

```
enigma [password]
```

You can include the password on the command line or enter it in the response to a prompt. The program returns an encrypted version that you can then enter in the options file.

Starting from Tivoli Workload Scheduler for Applications version 8.4, a new algorithm is applied to encrypt the password that is stored in the options file. When you use enigma, pwdcrypt, or the Option Editor the password is encrypted by applying this new algorithm. It is also applied if you open and save the options file with the Option Editor. The old password encryption is still recognized and processed by Tivoli Workload Scheduler for Applications version 8.4, so you do not need to encrypt the password again.

Defining Oracle E-Business Suite jobs

This chapter provides information for defining Oracle E-Business Suite jobs in Tivoli Workload Scheduler.

Defining jobs in Oracle E-Business Suite

In Oracle E-Business Suite, jobs that are to be scheduled with Tivoli Workload Scheduler are defined in the same way as other Oracle E-Business Suite jobs (and can be submitted in Oracle Concurrent Manager). When you are submitting the job in Oracle Concurrent Manager, specify the scheduling and resubmission information. When you are submitting the job through Tivoli Workload Scheduler, Tivoli Workload Scheduler will handle the scheduling and re-submission information.

To define a job, use the Define Concurrent Program Executables and Define Concurrent Program forms and fill in the general data information for the job. The job is now available for scheduling from Tivoli Workload Scheduler (and can also be submitted in Oracle Concurrent Manager).

Running a single job concurrently

If an Oracle E-Business Suite job is scheduled to run more than once in Tivoli Workload Scheduler, and Tivoli Workload Scheduler launches it concurrently, the jobs run normally. However, if the Oracle E-Business Suite user has specified that the job cannot run concurrently, the second job enters a state of WAIT until the first job completes.

Defining Oracle E-Business Suite jobs in Tivoli Workload Scheduler

Tivoli Workload Scheduler job definitions are required for each Oracle E-Business Suite job you plan to schedule. Oracle E-Business Suite jobs are defined in Tivoli Workload Scheduler like other jobs and include job name, user name, and recovery options. See "Defining jobs for supported agents" on page 36.

Task string parameters for Oracle E-Business Suite jobs

This section describes the task string parameters that manage the operation of Oracle E-Business Suite jobs. Specify them in the following places when you define their associated Tivoli Workload Scheduler jobs:

- In the **Task string** field of the Task page of the Properties - Job Definition panel, if you use the Dynamic Workload Console
- As arguments of the `scriptname` keyword in the job definition statement, if you use the Tivoli Workload Scheduler command line.
- As arguments of the `JOBCMD` keyword in the `JOBREC` statement in the `SCRIPTLIB` of Tivoli Workload Scheduler for z/OS, if you are scheduling in an end-to-end environment. The following is an example of a `JOBREC` statement:

```
JOBREC
  JOBCMD('/-user appslogin -job jobspecs')
  JOBUSR(TWS_user_name)
```

  where:

appslogin
The name of a valid Oracle Applications user (Apps login).

jobspecs
The attributes of the Oracle Applications job. For details, see Table 14.

TWS_user_name
The Tivoli Workload Scheduler for z/OS user who runs the `MCMAGENT` access method from the end-to-end scheduling environment.

Table 14 describes the parameters to define Oracle E-Business Suite jobs.

Table 14. Task string parameters for Oracle E-Business Suite jobs

Parameter	Description
`-user` *`appslogin`*	The name of a valid Oracle Applications user (Apps login).
`-job` *`jobspecs`*	The attributes of the Oracle Applications job. There are four job attributes that must be specified in the following order: 1. The name of the Applications module used to sign on to Oracle Applications (for example, `FND`). 2. A valid responsibility for the Applications module (for example, `'Application Developer'`). This parameter must be enclosed in single quotes. 3. The name of the application that registered the job (for example, `PO`). 4. The job name (for example, `POXACR`). The job name must be uppercase. For example: `-job FND 'Application Developer' PO POXACR`
`-prn` *`printopts`*	The two printer options for the job. They must be entered in the following order: 1. The printer name (optional). If omitted, the default printer specified in the options file is used. 2. The number of copies. For example: `-prn ps4 2`
tracelvl	Specify the trace setting for the job. Possible values are: **1** Only error messages are written in the trace file. This is the default. **2** Informational messages and warnings are also written in the trace file. **3** A most verbose debug output is written in the trace file. Refer to “Configuring the tracing utility” on page 281 for detailed information.

Table 14. Task string parameters for Oracle E-Business Suite jobs (continued)

Parameter	Description

`[-vn ('value'|null|:parameter)]`	The job's parameters, if any. This string is optional if no parameters are specified for the job in Oracle Applications.
	• *n* is the parameter number (v1, v2, and so on).
	• One of the following:
	value An actual value. Enclose *value* in single quotes if it contains special characters.
	:parameter The name of an already defined Tivoli Workload Scheduler parameter. For more information about Tivoli Workload Scheduler parameters, refer to the *Tivoli Workload Scheduler: User's Guide and Reference*.
	The access method automatically adds all the parameters that you do not specify in the definition with a value of `null`.
	These are two examples:
	`-v4 'abc'`
	`-v15 :p1`

The following is an example of an entire task string specification:

```
-user GLOBAL -job FND 'Application Developer' PO POXACR -prn ps4 2 -v1 'abc'
```

Tivoli Workload Scheduler and Oracle E-Business Suite job states

When an Oracle E-Business Suite job is launched by Tivoli Workload Scheduler, you can monitor its progress with **conman**. Table 15 provides you with a list of the state transitions in Tivoli Workload Scheduler and the corresponding Oracle E-Business Suite states.

Table 15. Tivoli Workload Scheduler and Oracle E-Business Suite job states

Oracle Concurrent Manager States	Tivoli Workload Scheduler States
Request Failure or Request not Found	UT (unsupported task)
Inactive	SUSP
Pending	WAIT
Pending Normal	ADD
Running Normal	EXEC
Completed Normal	SUCC
Completed Warning	SUCC
Completed Error	ABEND
Completed Terminated	ABEND
Completed Canceled	ABEND

For more information about job management, refer to *Tivoli Workload Scheduler: User's Guide and Reference*.

Part 3. PeopleSoft access method

Introducing Tivoli Workload Scheduler for Applications for PeopleSoft

What you need and what you can do with Tivoli Workload Scheduler for Applications for PeopleSoft.

Using Tivoli Workload Scheduler for Applications for PeopleSoft you can run and monitor PeopleSoft jobs from the Tivoli Workload Scheduler environment. These jobs can be run as part of a schedule or submitted for ad-hoc job processing. PeopleSoft extended agent or dynamic agent jobs can have all of the same dependencies and recovery options as other Tivoli Workload Scheduler jobs. PeopleSoft jobs must be defined in Tivoli Workload Scheduler to be run and managed in the Tivoli Workload Scheduler environment.

Software requirements

What you need to use Tivoli Workload Scheduler for Applications for PeopleSoft.

In addition to the supported operating systems and prerequisite software listed in "Installing, upgrading, and uninstalling IBM Tivoli Workload Scheduler for Applications" on page 3, you must have the following:

- PeopleTools version 8.44, 8.45, 8.46, 8.47, 8.48, 8.49, 8.50, or 8.51.
- Access to the psjoa.jar file located in the `*ps_home*\class` directory.

Features

Look at the tasks you can perform by using Tivoli Workload Scheduler for Applications for PeopleSoft.

Using Tivoli Workload Scheduler for Applications for PeopleSoft, you can perform the following tasks:

- Use Tivoli Workload Scheduler standard job dependencies on PeopleSoft jobs.
- Schedule PeopleSoft jobs to run on specified days, times, and in a prescribed order.
- Define inter-dependencies between PeopleSoft jobs and Tivoli Workload Scheduler jobs that run on different applications such as SAP R/3 and Oracle E-Business Suite.
- Define inter-dependencies between PeopleSoft jobs and jobs that run on different operating systems.

Roles and responsibilities

Here you can see the roles and responsibilities of all actors involved in the process model, and the tasks they perform.

In a typical enterprise, different users contribute to the implementation and operation of the product. Table 16 on page 60 describes the roles and responsibilities of all actors in the process model, showing the tasks they perform.

Table 16. Roles and responsibilities in Tivoli Workload Scheduler for Applications for PeopleSoft

User role	User task
Tivoli Workload Scheduler for Applications configurator	"Defining the configuration options" on page 61
Tivoli Workload Scheduler for Applications developer	• "Defining PeopleSoft jobs in Tivoli Workload Scheduler" on page 67 • "Configuring the job status mapping policy" on page 69
PeopleSoft administrator	• "Creating a batch processing ID in PeopleSoft" on page 64 • "Configuring the ITWS_PSXA PeopleSoft project" on page 64 • "Uploading the PeopleSoft project" on page 65

Scheduling process for the PeopleSoft supported agents

Tivoli Workload Scheduler can launch and monitor jobs in the PeopleSoft process scheduler using a PeopleSoft extended agent or dynamic agent workstation. The PeopleSoft supported agent (extended agent or dynamic agent) is defined in a standard Tivoli Workload Scheduler workstation definition. This definition is a logical workstation name and specifies the access method as `psagent`. The access method is used to communicate job requests to the PeopleSoft process scheduler.

To launch a PeopleSoft job, Tivoli Workload Scheduler runs the `psagent` method, passing it information about the job. An options file provides the method with the path, the executable, and other information about the PeopleSoft process scheduler and application server used to launch the job. The supported agent t can then access the PeopleSoft process request table and make an entry in the table to launch the job. Job progress and status information is written to the job's standard list file.

For extended agents, there is no need to install Database connectivity on fault-tolerant agents hosting PeopleSoft extended agents because the method currently uses the PeopleSoft 3-tier architecture. You must configure at least one PeopleSoft Application Server for the supported agent to work. The application server must be active to successfully submit jobs to the PeopleSoft process scheduler.

PeopleSoft job tracking in Tivoli Workload Scheduler

A PeopleSoft job is a collection of processes that run together as a single unit. Tivoli Workload Scheduler jobs can be defined in one of the following ways:

- As PeopleSoft jobs, that is, as a collection of PeopleSoft processes. In this case, the status of the PeopleSoft job is tracked, not the status of the individual processes within the job.
- As PeopleSoft processes. In this case, the status of the individual process is tracked and Tivoli Workload Scheduler schedules can be defined to create complex inter-dependencies and recovery options between PeopleSoft processes.

Security

Security for the PeopleSoft jobs is handled by standard Tivoli Workload Scheduler security.

Configuring the PeopleSoft access method

This chapter provides detailed reference information about the PeopleSoft options and how to define them in the options file.

Defining the configuration options

The Tivoli Workload Scheduler for Applications installation process creates a default global options file for the `psagent` access method, named `psagent.opts`. You can also create local files in the *`TWS_home`*`/methods` as following:

Extended agent

`XANAME``_psagent.opts` where *`XANAME`* is the extended agent workstation name.

Dynamic agent

`DYNAMIC_AGENT_FILE``_psagent.opts` where *DYNAMIC_AGENT_FILE* is any text string. This string does not necessarily correspond to the name of the dynamic agent workstation since the dynamic agent can have more than one .opts file associated. For more information, see "Setting options with the Option Editor" on page 26.

To edit both options file, you can use either the Option Editor available with this product, or any other text editor. For details about how to create and edit the options files with the Option Editor, see "Setting options with the Option Editor" on page 26. For examples of options files for this access method, see "PeopleSoft options file example" on page 63.

Table 17 describes the options for the `psagent` access method. Option names are case insensitive. Before you use a manually-created options file, check that all the option names are written correctly, otherwise they will be ignored.

Table 17. Psagent access method options

Option	Description
CHECKINTERVAL	(Optional) Specifies the frequency (in seconds) with which the **`psagent`** monitors a submitted process up to completion. The default is 120.
LJUSER	(Optional) Specifies the ID of the Tivoli Workload Scheduler user that runs the **`psagent`** to launch jobs (LJ tasks). This user must be a valid Tivoli Workload Scheduler user on the Tivoli Workload Scheduler hosting workstation.
PS_DISTSTATUS	(Optional) Determines whether the distribution status of the PeopleSoft request is taken into account when determining the status of the Tivoli Workload Scheduler job. Values are *0* (not taken into account) or *1* (taken into account - the default value).
PSFT_OPERATOR_ID	(Mandatory) Specifies the PeopleSoft operator ID used for the connection to the PeopleSoft application server.
PSFT_OPERATOR_PWD	(Mandatory) Specifies the encrypted password (case-sensitive) of the PeopleSoft operator ID used for the connection to the PeopleSoft application server. For details about how to encrypt the password, see "Encrypting PeopleSoft operator passwords" on page 63.

Table 17. Psagent access method options (continued)

Option	Description
PSJOAPATH	(Optional) Specifies the full path name of the psjoa.jar file. If this option is not set, the path name provided during the installation is used. Ensure that you identify the version of the psjoa.jar file that corresponds to the version of PeopleSoft that you are using.
RUNLOCATION	(Optional) Specifies the default PeopleTools process server that processes the requests.
SERVER_NAME_LIST	(Mandatory) Specifies the list of application servers that the **psagent** tries to connect to. It is a comma-separated list of addresses in the format: `server:port [,server:port] ...` where: *server* Specifies the host name or TCP/IP address of the server *port* Specifies the port number the server is listening on.
TWS_MAX_WAIT_TIME	(Optional) Specifies the maximum time that the supported agent waits (timeout) after a failed operation on the PeopleSoft application server before retrying the operation. The default is 10 seconds.
TWS_MIN_WAIT_TIME	(Optional) Specifies the minimum time that the supported agent waits (timeout) after a failed operation on the PeopleSoft application server before retrying the operation. The default is 5 seconds.
TWS_RETRY	(Optional) The maximum number of times that the supported agent attempts a failed operation on the PeopleSoft application server. The default is 5.
TWSXA_INLINE_CI	(Optional) Specifies the name of the component interface that the psagent invokes to submit jobs to PeopleSoft. The default is ITWS_PROCESSREQUEST. If you use this default, you must perform the customization steps described in "Configuring the ITWS_PSXA PeopleSoft project" on page 64. If you do not plan to schedule jobs containing in-line variables, and you do not want to perform the additional customization steps, you must replace the default value with PROCESSREQUEST. This is the component interface invoked by previous versions of the access method; it does not allow the use of in-line variables.
TWSA_SCHED_METH	(Optional) Specifies the name of the PeopleSoft method invoked by the component interface specified in TWSXA_INLINE_CI. Both ITWS_PROCESSREQUEST and PROCESSREQUEST use the default method Schedule. If you are using either of these component interfaces, leave the default. If you are using a different component interface, specify the name of the method called by your component interface, respecting the case of the PeopleSoft object name.

PeopleSoft options file example

Below is a sample options file for PeopleTools version 8.44. It can help you determine your specific site requirements although your options file might be different.

Remember to save the file in the *TWS_home*\methods directory.

```
LJuser=TwsUsr
CheckInterval=120
PSFT_OPERATOR_ID=PSHC
PSFT_OPERATOR_PWD=*****
SERVER_NAME_LIST=9.87.120.36:9000
```

If you create the options file manually, you must encrypt the PeopleSoft operator password, as described in “Encrypting PeopleSoft operator passwords.”

Encrypting PeopleSoft operator passwords: When you add or change the PeopleSoft operator password using the Option Editor, the password is automatically encrypted and inserted into the file, and appears as asterisks on the screen.

When you add or change the PeopleSoft user password using a text editor, you must run the **pwdcrypt** program to encrypt the password before writing it in the file. To run the encryption program, enter the following command:

```
pwdcrypt password
```

The program returns an encrypted version that you can then copy and paste into the options file.

With Tivoli Workload Scheduler for Applications version 8.4, a new algorithm is applied to encrypt the password that is stored in the options file. When you use enigma, pwdcrypt, or the Option Editor the password is encrypted by applying this new algorithm. It is also applied if you open and save the options file with the Option Editor. The old password encryption is still recognized and processed by Tivoli Workload Scheduler for Applications version 8.4, so you do not need to encrypt the password again.

Connecting to more than one PeopleSoft application server

It might be necessary for thepsagent method to connect to more than one PeopleSoft application server. For example, a single installation of PeopleSoft might have a TEST, DEMO, and PRODUCTION environment, each with a separate application server. This requires that the psagent method uses a separate connect string for each application server.

To support this, you can set up multiple PeopleSoft extended agent workstations that connect to the same method but use different options files. When a workstation starts the method, it first looks for the options file with extended agent workstation name prepended to psagent.opts. For example, a PeopleSoft extended agent named ps847system would have the following options file:

```
PS847SYSTEM_psagent.opts
```

The **psagent** method searches first for an options file with the extended agent workstation name, and then for the default psagent.opts file. This allows the user to set up an extended agent for each PeopleSoft application server.

To connect to only one application server, use the default name for the options file, **psagent.opts**.

Creating a batch processing ID in PeopleSoft

Create an operator ID in PeopleSoft dedicated to batch scheduling. This operator ID must be granted authority to use the Component Interface in the PeopleTools environment. All the jobs submitted by IBM Tivoli Workload Scheduler should use this operator ID.

Configuring the ITWS_PSXA PeopleSoft project

The configuration steps described in this section are necessary to enable Tivoli Workload Scheduler for Applications to schedule PeopleSoft jobs that have in-line variables in their definitions.

The `ITWS_PROCESSREQUEST` component interface works around some limitations of the PeopleSoft APIs when invoked from a batch environment. Because of these limitations, Tivoli Workload Scheduler cannot schedule jobs defined with in-line bind variables. With current PeopleSoft APIs, data that is stored in the PeopleSoft database and referred to by a `runcontrol ID parameter` that is used to retrieve a `runcontrol data record`, needs to be loaded into the `Component Buffer` before scheduling the API invocation. This cannot be done from a batch environment. Therefore, when invoking the PeopleSoft scheduling APIs from a batch interface, the data related to the `runcontrol ID` is not available for the submission of a job, even though it is available in the database. When unresolved data is present in the submitted job, the PeopleSoft system refuses submission and ends with an error.

The `ITWS_PROCESSREQUEST` component interface enables Tivoli Workload Scheduler for Applications to schedule PeopleSoft jobs that have in-line variables in their definitions. By invoking this component interface, the access method provides the ability to use data stored in the PeopleSoft database to resolve in-line variable values by taking data from the database and substituting it with variable definitions. It then allows job submission regardless of the use of in-line variable definitions in the jobs. The variable substitution mechanism does not support `work` records, so if the PeopleSoft process uses work records in its parameter list, you find a message similar to the following in the Tivoli Workload Scheduler joblog:

```
Error Position:  21
Return:  942 - ORA-00942: table or view does not exist
Statement:
select nvsdlist from PS_NVS_WRK WHERE BUSINESS_UNIT = :1 AND REPORT_ID = :2
Original Statement:
SELECT NVSDLIST FROM PS_NVS_WRK WHERE BUSINESS_UNIT = :1 AND REPORT_ID = :2.
```

To identify work records, use the following PeopleSoft naming conventions:
- A derived work record name ends with '_WRK'.
- A work record definition name for Structured Query Report reports starts with R_

When you use Tivoli Workload Scheduler to submit a process that has in-line bind variables, the name of the process type in the PeopleSoft GUI becomes `ITWS_`*`process type`*. For example, `SQR Process` becomes `ITWS_SQR Process`.

To schedule a job that contains in-line variables in its definition you must perform the following tasks:

- Leave the value of the `TWSXA_INLINE_CI` option set to `ITWS_PROCESSREQUEST`, that is the default value. See "Defining the configuration options" on page 61 for a detailed explanation.
- Upload the PeopleSoft project as described in "Uploading the PeopleSoft project" on page 65.

Uploading the PeopleSoft project

This section describes how to upload a new PeopleSoft project related to PeopleTools 8.44, or later, into the PeopleSoft database. The name of the PeopleSoft project is ITWS.

After installing the product as described in "Installing, upgrading, and uninstalling IBM Tivoli Workload Scheduler for Applications" on page 3, perform the following steps:

1. Mount the PT844 PeopleSoft project directory or copy it to the workstation from where you launch the Application Designer. Tivoli Workload Scheduler installs the PeopleSoft project directories, as shown in the following structure:

   ```
   TWS_HOME\methods\
   \---PeopleSoft
           \---PT844
            \---ITWS_PSXA
                    ITWS_PSXA.ini
                    ITWS_PSXA.XML
   ```

2. Start the Application Designer and from the sign-on window select to start the Application Designer in tier-two mode by entering the following information:
   - **Connection Type**: *database used*; for example, Oracle
   - **Database Name**: *database instance name*
   - **User ID**: *PeopleSoft operator name*; for example, PS
   - Password of user ID
3. Using the Application Designer, select **Tools** -> **Copy Project**-> **From file...**
4. Using the browser, edit the full path to specify the folder where the project that you want to load is located.

   The project is contained in the *TWS_home*/methods/PeopleSoft/PT844 subdirectories.

   After you specify the project folder, a list of projects appears in the **Project Name** field of the Copy Project From File window.
5. Choose ITWS_PSXA and click **Open**. If you already configured ITWS_PSXA (perhaps after installing a fix pack), a confirmation window enquires if you want to replace the existing one. Click **Yes**.

   The Copy window is displayed showing a list of definition types.
6. Click **Options** to select the new settings.
   a. Click **Report Filter**
   b. Click **Select All**
   c. Click **OK**
   d. Click **Select All**
   e. Click **Copy**. A progress bar is displayed.

   After loading the project, the PeopleSoft Database contains the following objects:
   - ITWS process type definitions
   - ITWS permissions list
   - ITWS component interfaces
7. Create the ITWS_ROLE security role. You can use either the PeopleSoft Web GUI or the Application Designer. Follow the steps below:

   From the menu of the PeopleSoft Web GUI:
   a. Select: **People tools** -> **Security** -> **Permission and Roles** -> **Roles**
   b. Select the **Add a new value** tab

   c. Type or select ITWS_ROLE in the **Role Name** field
   d. Select the **Permissions list** tab -> **ITWS** -> **Save**

   From the Application Designer GUI:

   a. Using Maintain Security, edit the ITWS_ROLE window
   b. Select the **Permissions list** tab -> **ITWS** -> **Save**

8. Grant ITWS_ROLE authority to all users who want to schedule jobs from Tivoli Workload Scheduler. You can use either the PeopleSoft Web GUI or the Application Designer. Follow the steps below:

   From the PeopleSoft Web GUI:

   a. Select: **People tools** -> **Security** -> **User Profiles**
   b. Type the user name of the user who wants to schedule jobs from Tivoli Workload Scheduler
   c. Select the **Roles** tab
   d. Add ITWS_ROLE and save

   From the Application Designer GUI:

   a. Using Maintain Security, edit the user name
   b. Select the **Roles** tab
   c. Add ITWS_ROLE and save

9. Add the ITWS process type definitions to the required PeopleTools process scheduler. You can use either the PeopleSoft Web GUI or the Application Designer. Follow the steps below:

   From the PeopleSoft Web GUI:

   a. Select **PeopleTools** -> **Process Scheduler** -> **Servers**
   b. Select your PeopleTools server
   c. Add the ITWS_* Type definitions and save

   From the Application Designer GUI:

   a. Select **Process Scheduler Manager**
   b. Select your PeopleTools server
   c. Add the ITWS_* Type definitions and save

   Note: From the SQL interactive command line, the same task can be performed by the following sample statement, customized for your database environment:

   ```
   INSERT INTO PS_SERVERCLASS SELECT o.SERVERNAME,
   o.OPSYS,'ITWS_'||o.PRCSTYPE,o.PRCSPRIORITY,
   o.MAXCONCURRENT FROM PS_SERVERCLASS
   o WHERE (  SELECT count(*) FROM PS_SERVERCLASS i WHERE
   i.SERVERNAME=o.SERVERNAME AND   i.OPSYS=o.OPSYS AND
   i.PRCSTYPE='ITWS_'||o.PRCSTYPE ) = 0
   AND ( select count(*) from PS_PRCSTYPEDEFN
   a where a.PRCSTYPE='ITWS_'||o.PRCSTYPE AND a.OPSYS=o.OPSYS ) > 0
   ```

10. Restart the process servers.

You do not need to change the existing Tivoli Workload Scheduler job definitions, except for the scheduling nVision process, where the runcontrol ID must be specified using the *BUSINESS_UNIT.REPORT_ID* convention.

The following is an example of a job definition for the scheduling nVision process:

```
-process 'NVSRUN' -type nVision-Report -runcontrol AUS01.VARIABLE
```

where NVSRUN is the process name and AUS01.VARIABLE is the *BUSINESS_UNIT.REPORT_ID*.

Defining PeopleSoft jobs

This chapter provides job definition information for jobs using the extended agent for PeopleSoft.

Defining PeopleSoft jobs in Tivoli Workload Scheduler

A Tivoli Workload Scheduler job definition is required for every PeopleSoft job you want to manage. A Tivoli Workload Scheduler job is associated to an already defined PeopleSoft job and its definition includes:

- The name of the Tivoli Workload Scheduler job that runs the PeopleSoft job
- The name of the extended agent workstation or workstation class where the Tivoli Workload Scheduler job runs
- The name of the user launching the job
- Recovery options
- The Script file specifications

For more information, refer to “Defining jobs for supported agents” on page 36.

Task string parameters for PeopleSoft jobs

This section describes the task string parameters that control the operation of PeopleSoft jobs. You must specify them in the following places when you define their associated Tivoli Workload Scheduler jobs:

- In the **Task string** field of the Task page of the Properties - Job Definition panel, if you use the Dynamic Workload Console
- As arguments of the scriptname keyword in the job definition statement, if you use the Tivoli Workload Scheduler command line.
- As arguments of the JOBCMD keyword in the JOBREC statement in the SCRIPTLIB of Tivoli Workload Scheduler for z/OS, if you are scheduling in an end-to-end environment. In this case the entire string following the JOBCMD keyword must be enclosed within quotation marks (").

  The following is an example of a JOBREC statement:

```
JOBREC
  JOBCMD("/-process process_name -type 'process_type' -runcontrol runcontrol_ID")
  JOBUSR(TWS_user_name)
```

  where:

  process_name
  The process name for the PeopleSoft job.

  process_type
  The process type for the PeopleSoft job. This entry must be enclosed within single quotes.

  runcontrol_ID
  The runcontrol ID for the PeopleSoft job.

  TWS_user_name
  The Tivoli Workload Scheduler for z/OS user who runs the psagent access method from the end-to-end scheduling environment.

Table 18 on page 68 describes the parameters to define PeopleSoft jobs.

Table 18. Task string parameters for PeopleSoft jobs

Parameter	Description
-process	The process name for the PeopleSoft job.
-type	The process type for the PeopleSoft job. This entry must be enclosed within single quotes.
-runcontrol	The runcontrol ID for the PeopleSoft job.
-outputdest	The destination of the PeopleSoft job output.
-outputtype	The output type of the PeopleSoft job. Possible values are: • Any • Email • File • None • Printer • Web • Window If you do not specify any value, Tivoli Workload Scheduler uses the value associated to the PeopleSoft job you are submitting. **Note:** Depending on the PeopleSoft configuration, some combinations of the value of this option with the value of the **outputformat** option are not supported. In this case the PeopleSoft default value is used.
-outputformat	The output format of the PeopleSoft job. Valid values are: **None** PDF **CSV** PS **DOC** RPT **Default** RTF **HP** SPF **HTM** TXT **LP** WKS **OTHER** XLS **Note:** Depending on the PeopleSoft configuration, some combinations of the value of this option with the value of the **outputtype** option are not supported. In this case the PeopleSoft default value is used.
-runlocationdescr	The PeopleSoft process scheduler responsible for processing the PeopleSoft job.
-foldername	The name of the report folder used for this job. The folder must have been already created using PeopleSoft Report Manager.
tracelvl	Specify the trace setting for the job. Possible values are: **1** Only error messages are written in the trace file. This is the default. **2** Informational messages and warnings are also written in the trace file. **3** A most verbose debug output is written in the trace file. Refer to "Configuring the tracing utility" on page 281 for detailed information.

Note: No syntax checking is performed on the output control values (`outputdest`, `outputtype`, `outputformat`, and `foldername`). If the values are not recognized, defaults are used.

The following is an example of a task string specification for a PeopleSoft 8.44 job:

```
-process XRFWIN -type 'SQR Report' -runcontrol 1 -runlocationdescr PSNT
```

Configuring the job status mapping policy

Tivoli Workload Scheduler calculates the status of a Tivoli Workload Scheduler job based on the PeopleSoft job Run Status and Distribution Status. In PeopleSoft, the run status monitors the running of the job until it reaches a final status; the distribution status monitors the status of the output of the job. If the final status of a PeopleSoft job is neither success nor warning, Tivoli Workload Scheduler ignores the distribution status and the Tivoli Workload Scheduler job status is ABEND.

If the final status of a PeopleSoft job is success or warning, you can decide whether to use the distribution status of the PeopleSoft job when determining the status of the Tivoli Workload Scheduler job by setting the PS_DISTSTATUS option in the options file:

0 The distribution status is ignored and the Tivoli Workload Scheduler job status is calculated as shown in Table 20 on page 70.

1 The distribution status is used and the Tivoli Workload Scheduler job status is calculated as shown in Table 19. This is the default value.

Table 19 shows the relationship between the run status, the distribution status, and the Tivoli Workload Scheduler job status. The return code associated with the status is shown in parentheses. Tivoli Workload Scheduler uses this return code to evaluate the return code condition you specified in the **Return Code Mapping Expression** field in the Properties panel of the job definition. For more details about this field, refer to the online help by clicking the "?" in the top-right corner of the panel.

Table 19. Relationship between the run status, the distribution status, and the Tivoli Workload Scheduler job status

PeopleSoft job run status	PeopleSoft job distribution status	Tivoli Workload Scheduler job status
• Success (9) • Warning (17)	• Posted (5) • None (0)	SUCC
• Success (9) • Warning (17)	• Not Posted (4) • Delete (6)	ABEND
• Success (9) • Warning (17)	• Not Available (1) • Processing (2) • Generated (3) • Posting (7)	EXEC
• Cancel (1) • Delete (2) • Error (3) • Canceled (8) • No Success (10) • Blocked (18) • Restart (19)	Any distribution status	ABEND

Table 20 on page 70 shows the relationship between the PeopleSoft run status and the Tivoli Workload Scheduler job status. The return code associated with the status is shown in parentheses. Tivoli Workload Scheduler uses this return code to evaluate the return code condition you specified in the **Return Code Mapping**

Expression field in the Properties panel of the job definition. For more details about this field, refer to the online help by clicking the "?" in the top-right corner of the panel.

Table 20. Relationship between the run status and the Tivoli Workload Scheduler job status

PeopleSoft final run status	IBM Tivoli Workload Scheduler status
Cancel (1)	ABEND
Delete (2)	ABEND
Error (3)	ABEND
Hold (4)	WAIT
Queued (5)	WAIT
Initiated (6)	INIT
Processing (7)	EXEC
Canceled (8)	ABEND
Success (9)	SUCC
No Success (10)	ABEND
Pending (16)	EXEC
Warning (17)	SUCC
Blocked (18)	ABEND
Restart (19)	ABEND

Note: If Tivoli Workload Scheduler fails to retrieve the status of the PeopleSoft job, the Tivoli Workload Scheduler job status is **DONE**.

Part 4. SAP R/3 access method

Introducing Tivoli Workload Scheduler for Applications for SAP

Using Tivoli Workload Scheduler for Applications for SAP, you can create, schedule, and control SAP jobs using the job scheduling features of Tivoli Workload Scheduler.

Tivoli Workload Scheduler for Applications Version 8.6 is certified by SAP for the following SAP interfaces:

- BC-XBP 6.10 (V2.0) - Background Processing
- BC-XBP 7.00 (V3.0) - Background Processing
- BW-SCH 3.0 - Business Information Warehouse

Certification Category: Background Processing, Business Information Warehouse, Job Scheduling, Platform User Licensing Compliant, Scheduling, Solution Manager Ready, System Management

SAP Certified - Integration with SAP NetWeaver.

Note: For detailed information, see the SAP online product partner directory.

Software requirements

For the most up-to-date information on system requirements for Tivoli Workload Scheduler for Applications, refer to the *Detailed System Requirements*.

In addition to the supported operating systems and prerequisite software listed in "Installing, upgrading, and uninstalling IBM Tivoli Workload Scheduler for Applications" on page 3, you must have the following:

- SAP R/3 version 6.10, 6.20, 6.30, 6.40, 7.00, 7.01, 7.02, 7.10, 7.11, 7.20, 7.30, or 7.31.
- The appropriate SAP R/3 Support Packages.

Table 21. SAP R/3 support packages

SAP Basis Versions	SAP Notes
6.10 unicode/non-unicode	357924, 399449, 435111, 453135, 517672, 602683, 604496, 610480, 612564, 635095, 684106, 758829, 784446, 790615, 792767, 833777
6.20 unicode/non-unicode	602683, 604496, 612564, 635095, 684106, 758829, 792767, 833777, 1079601, 1257432, 1299738
6.30 unicode/non-unicode	602683, 604496, 612564, 684106, 758829, 792767, 833777, 1299738
6.40 unicode/non-unicode	602683, 604496, 612564, 635095, 684106, 758829, 784446, 790615, 792767, 833777, 1079601, 1257432, 1299738
7.00 unicode/non-unicode	684106, 792767, 833777, 862989, 1257432, 1299738
7.01 unicode/non-unicode	1658978, 1402400, 1447880, 1461421, 1639210
7.02 unicode/non-unicode	786412, 1658978, 1402400, 1447880, 1461421, 1639210

Table 21. SAP R/3 support packages (continued)

SAP Basis Versions	SAP Notes
7.10 unicode/non-unicode	729317, 1257432, 1299738
7.20 unicode/non-unicode	786412, 1658978, 1402400, 1447880, 1461421, 1639210
7.30 unicode/non-unicode	1658978, 1447880, 1639210
7.31 unicode/non-unicode	1658978, 1447880, 1639210

Note:

1. Note 399449 resolves print parameter issues. Notes 357924 and 517672 are release-independent.
2. Note 758829 resolves job step management problems. See Table 88 on page 286.
3. Notes 602683, 604496, and 612564 are required if you want to use the BC-XBP 2.0 interface.
4. Note 684106 applies only to Windows operating systems.

InfoPackages are supported by Tivoli Workload Scheduler for Applications on SAP Business Warehouse systems 2.0B, or later.

Process chains are supported on SAP Business Warehouse systems 3.0B, or later. The Support Package 9 (SAPKW31009) for SAP Business Warehouse Version 3.1 is required for the extended agent to be able to launch process chains.

Ensure that you have installed the required SAP Business Warehouse Support Package as follows:

Table 22. SAP Business Warehouse support packages

SAP Business Warehouse Versions	SAP Notes
3.0B, 3.50	786748, 786749, 795858, 826878, 1059049
3.10	786748, 786749, 795858, 826878
7.00	826878, 1059049, 1049735, 1100457, 1130034, 1723482
7.01	1723482
7.02	1723482
7.10	1723482
7.11	1723482
7.20	1723482
7.30	1723482
7.31	1723482

To have the SAP R/3 access method work correctly, ensure that you have installed the required SAP Remote Function Call Libraries, as shown in Table 23. You must download these libraries from the SAP website http://service.sap.com/swdc and copy them to the *TWS_home*\methods directory.

Table 23. SAP Remote Function Call libraries

SAP Remote Function Call Library Versions	Operating System
SAP RFC SDK 6.40	32-bit AIX operating systems

Table 23. SAP Remote Function Call libraries (continued)

SAP Remote Function Call Library Versions	Operating System
SAP RFC SDK 6.40	32-bit HP-UX 11 on PA-RISC systems
SAP RFC SDK 6.40	32-bit Solaris on SPARC systems
SAP RFC SDK 6.20	32-bit Linux on zSeries systems
SAP RFC SDK 7.10	All other supported platforms **Important:** On 32-bit and 64-bit Linux xSeries systems, use the 32-bit unicode RFC SDK.

To find the current version of the Remote Function Call library, see SAP note 413708.

To enable Distributed Statistical Records (DSR), you must first install the SAP DSR library; see http://help.sap.com/saphelp_nw04/helpdata/en/44/2c6f553f9c4966e10000000a11466f/content.htm. For more information about enabling DSR in ITS, see SAP Note 654849.

Server groups are supported by Tivoli Workload Scheduler for Applications on the following versions of SAP Basis:

Table 24. Supported SAP Basis versions and support packages

SAP Basis Version	Support Package
6.10	40
6.20	41
6.40	30
7.00	27
7.01	11
7.02	11
7.10	15
7.11	09
7.20	07
7.30	07
7.31	03

For a list of functions and limitations for SAP R/3 server groups, refer to SAP Notes 786412, 729317, 612838, 1658978, and 1402400.

The following SAP notes are prerequisites for r3batch.

Modify job header, Job class missing (SAP Note 1447880)

Table 25. Supported SAP Basis versions and support packages

SAP Basis Version	Support Package
7.00	26
7.01	11
7.02	10
7.10	13
7.11	08

Table 25. Supported SAP Basis versions and support packages (continued)

SAP Basis Version	Support Package
7.20	06
7.30	05
7.31	01

Modify SAP variant, Variant type missing (SAP Note 1461421)

Table 26. Supported SAP Basis versions and support packages

SAP Basis Version	Support Package
7.00	22
7.01	07
7.02	05
7.10	11
7.11	06
7.20	04

Modify SAP variant, Variant attributes overwritten (SAP Note 1639210)

Table 27. Supported SAP Basis versions and support packages

SAP Basis Version	Support Package
7.00	26
7.01	11
7.02	11
7.10	14
7.11	09
7.20	07
7.30	07
7.31	02

Features

Table 28 shows the tasks you can perform with Tivoli Workload Scheduler for Applications for SAP either in a distributed or an end-to-end environment.

Table 28. Tivoli Workload Scheduler for Applications for SAP features

Feature	Distributed environment	End-to-end
Using Tivoli Workload Scheduler standard job dependencies and controls on SAP jobs	✓	✓
Listing jobs, defining jobs, variants, and extended variants using the Tivoli Workload Scheduler interface	✓	✓
Defining jobs and variants dynamically at run time	✓	✓
Scheduling SAP jobs to run on specified days and times, and in a prescribed order	✓	✓
Defining the national language support options	✓	✓

Table 28. Tivoli Workload Scheduler for Applications for SAP features (continued)

Feature	Distributed environment	End-to-end
Using the SAP Business Warehouse Support functions	✓	✓
Customizing job execution return codes	✓	✓
Using SAP logon groups for load balancing and fault-tolerance	✓	✓
Using Business Component-eXternal Interface Background Processing (XBP 2.0 and later) interface support to:	• Collect intercepted jobs. • Track child jobs • Keep all job attributes when you rerun a job • Raise events	• Track child jobs • Keep all job attributes when you rerun a job • Raise events
Using Business Component-eXternal Interface Background Processing (XBP 3.0) interface support to:	• Create criteria profiles to log raised events, reorganize the event history, and intercept and relaunch jobs, according to the criteria you specify. • SAP application log and application return code • Spool list request and display for jobs that have run. • Temporary variants	• Create criteria profiles to log raised events, reorganize the event history, and intercept and relaunch jobs, according to the criteria you specify. • SAP application log and application return code • Spool list request and display for jobs that have run. • Temporary variants
Assigning an SAP job to a server group, for batch processing	✓	✓
Exporting SAP factory calendars and adding their definitions to the Tivoli Workload Scheduler database	✓	
Defining internetwork dependencies and event rules for Tivoli Workload Scheduler based on SAP events	✓	
Defining event rules based on IDoc records	✓	
Defining event rules based on CCMS Monitoring Architecture alerts	✓	
Rerunning a job that submits a process chain from a specific process, from failed processes, or as a new instance	✓	✓
Displaying the details of a job that submits a process chain	✓	✓
Enabling job throttling	✓	✓

Scheduling process for the agent workstation hosting the r3batch access method

Tivoli Workload Scheduler launches jobs in SAP using Tivoli Workload Scheduler jobs defined to run on a supported agent workstations.

Supported agent workstations include:

- extended agents
- dynamic agents
- Tivoli Workload Scheduler for z/OS agents

See "Supported agent workstations" on page 2 for more details about these agent workstations.

The supported agent workstations use the access method, `r3batch`, to pass SAP job-specific information to predefined SAP instances. The access method uses information provided in an options file to connect and launch jobs on an SAP instance.

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

On dynamic agent workstations, more than one options file can be associated to the workstation.

For more information about job management, refer to the *Tivoli Workload Scheduler: User's Guide and Reference*.

For more detailed information about configuration files on extended agents and dynamic agents, see "Configuring the SAP R/3 access method" on page 90.

Roles and responsibilities

In a typical enterprise, different users contribute to the implementation and operation of the product. Table 29 describes the roles and responsibilities of all users in the process model, showing the tasks they perform.

Table 29. Roles and responsibilities in Tivoli Workload Scheduler for Applications for SAP

User role	User task
Tivoli Workload Scheduler for Applications administrator	• "Creating the IBM Tivoli Workload Scheduler RFC user" on page 80 • "Creating the authorization profile for the Tivoli Workload Scheduler user" on page 80 • "Copying the correction and transport files" on page 83 • "Importing ABAP/4 function modules into SAP R/3" on page 84

Table 29. Roles and responsibilities in Tivoli Workload Scheduler for Applications for SAP (continued)

User role	User task
Tivoli Workload Scheduler for Applications configurator	• "Changing the Tivoli Workload Scheduler RFC user ID password" on page 87 • "Migrating from previous versions" on page 88 • "Print parameter and job class issues" on page 89 • "Defining the configuration options" on page 91 • "Connecting to the SAP system" on page 103 • "Creating SAP Standard R/3 jobs from the Dynamic Workload Console" on page 106 • "Using the BDC Wait option" on page 149 • "Implementing job interception" on page 150 • "Defining user authorizations to manage SAP R/3 Business Warehouse InfoPackages and process chains" on page 159 • "Setting and using job throttling" on page 174 • "Exporting SAP R/3 factory calendars" on page 178 • "Setting National Language support options" on page 207
Tivoli Workload Scheduler for Applications developer	• "Editing a standard SAP job" on page 114 • "Task string to define SAP jobs" on page 115 • "Displaying details about a standard SAP job" on page 123 • "Verifying the status of a standard SAP job" on page 124 • "Deleting a standard SAP job from the SAP database" on page 124 • "Balancing SAP workload using server groups" on page 125 • "Defining SAP jobs dynamically" on page 131 • "Managing SAP R/3 Business Warehouse InfoPackages and process chains" on page 159
Tivoli Workload Scheduler for Applications developer	• "Defining internetwork dependencies and event rules based on SAP R/3 background events" on page 181 • "Defining event rules based on IDoc records" on page 189 • "Defining event rules based on CCMS Monitoring Architecture alerts" on page 197
Tivoli Workload Scheduler for Applications operator	• "Rerunning a standard SAP job" on page 129 • "Mapping between Tivoli Workload Scheduler and SAP job states" on page 126 • "Raising an SAP event" on page 127 • "Killing an SAP job instance" on page 127 • "Displaying details about a process chain job" on page 164

Configuring user authorization (Security file)

Tivoli Workload Scheduler manages security through the use of a configuration file, the security file. In the security file, you specify which scheduling objects a user can manage and how. You define these settings by writing user definitions. A user definition is an association between a name and a set of users, the objects they can access, and the actions they can perform on the specified objects.

For more detailed information about the security file, security file syntax, and how to configure the security file, refer to Configuring user authorization (Security file).

The following table displays the access keywords required to grant authorization to access and work with SAP scheduling objects assigned to Tivoli Workload Scheduler users.

Table 30. Access keywords for activities with SAP scheduling objects

Activity		**Access keywords required**
Dynamic Workload Console	Define or search for SAP jobs on an extended agent workstation.	display on the workstation
	Retrieve the spool list on an extended agent workstation.	display on the workstation
	Rerun from a step on an extended agent.	rerun on the job
	Define or search for SAP jobs on a dynamic agent workstation, pool, or dynamic pool.	display and run on the workstation
	Retrieve the spool list on a dynamic agent workstation, pool, or dynamic pool.	display and run on the job
	Rerun from a step on a dynamic agent workstation, pool, or dynamic pool.	rerun on the job

Unicode support

Tivoli Workload Scheduler for Applications for SAP supports the Unicode standard.

What is Unicode

Unicode was devised to address the problem caused by the profusion of code sets. Since the early days of computer programming hundreds of encodings have been developed, each for small groups of languages and special purposes. As a result, the interpretation of text, input, sorting, display, and storage depends on the knowledge of all the different types of character sets and their encodings. Programs are written to either handle one single encoding at a time and switch between them, or to convert between external and internal encodings.

The problem is that there is no single, authoritative source of precise definitions of many of the encodings and their names. Transferring text from one computer to another often causes some loss of information. Also, if a program has the code and the data to perform conversion between many subsets of traditional encodings, then it needs to hold several Megabytes of data.

Unicode provides a single character set that covers the languages of the world, and a small number of machine-friendly encoding forms and schemes to fit the needs of existing applications and protocols. It is designed for best interoperability with both ASCII and ISO-8859-1, the most widely used character sets, to make it easier for Unicode to be used in applications and protocols.

Unicode makes it possible to access and manipulate characters by unique numbers, their Unicode code points, and use older encodings only for input and output, if at all. The most widely used forms of Unicode are:

- UTF-32, with 32-bit code units, each storing a single code point. It is the most appropriate for encoding single characters.
- UTF-16, with one or two 16-bit code units for each code point. It is the default encoding for Unicode.
- UTF-8, with one to four 8-bit code units (bytes) for each code point. It is used mainly as a direct replacement for older MBCS (multiple byte character set) encodings.

Unicode support on r3batch

Starting with R/3 version 4.7 (R/3 Enterprise), Unicode is used on all layers of the R/3 system:

- UTF-8, UTF-16, and UTF-32 on the database
- UTF-16 on the application server and graphical user interface

`r3batch` uses the UTF-8 code page internally. Because it communicates with SAP R/3 at the application server layer, it uses UTF-16 when communicating with Unicode-enabled SAP R/3 systems.

To use Unicode support, the following conditions must be met:

- Tivoli Workload Scheduler for Applications for SAP must run on one of the following operating systems:
  - Microsoft Windows Server 2003 on IA32, 32-bit
  - Microsoft Windows Server 2003 on x64, 64-bit
  - Microsoft Windows Server 2003 on IA64, 64-bit
  - IBM AIX, 64-bit
  - Sun Solaris Operating Environment, 64-bit
  - Sun Solaris Operating Environment for Opteron
  - HP-UX for Integrity
  - HP-UX for PA-RISC, 64-bit
  - Linux on xSeries®
  - Linux on zSeries, 64-bit

  The product does not support Unicode on the other operating systems where it can be installed.
- The SAP R/3 systems that communicate with `r3batch` must be running Unicode-enabled SAP R/3 versions.

If these conditions are not met, you cannot use Unicode support and must make sure that `r3batch`, the Dynamic Workload Console, and the target SAP R/3 system code page settings are aligned. Use the options related to national language support described in “SAP R/3 supported code pages” on page 208.

Configuring the SAP R/3 environment

To communicate and manage the running of jobs on SAP R/3 systems using the access method for SAP R/3, perform the following configuration steps in the SAP R/3 environment.

The steps described in this chapter require that you have knowledge of an SAP R/3 Basis Administrator.

Overview

Here is an overview of the customization procedure:

1. Create a new user ID for RFC communications in SAP R/3 for Tivoli Workload Scheduler.
2. Create the authorization profile as described in "Creating the authorization profile for the Tivoli Workload Scheduler user."
3. Copy the correction and transport files from the Tivoli Workload Scheduler server to the SAP R/3 server.
4. Import the correction and transport files into SAP R/3 and verify the installation.

Note: The import procedure adds new ABAP/4 function modules and several new internal tables to the SAP R/3 system. It does not modify any of the existing objects.

Creating the IBM Tivoli Workload Scheduler RFC user

For Tivoli Workload Scheduler to communicate with SAP R/3, you must create a user ID in SAP R/3 for Tivoli Workload Scheduler batch processing. For security reasons, use a new user ID rather than an existing one.

1. Create a new RFC user ID.
2. Give this new RFC user ID the following attributes:
   - A user type of **CPIC**, **Communications**, or **DIALOG**, depending on the SAP R/3 release.
   - A password at least six characters in length. Tivoli Workload Scheduler requires this password to start or monitor SAP R/3 jobs. If this password changes in SAP R/3, you must update the options file used by `r3batch` with the new password.
   - The appropriate security profiles, depending on your version of SAP R/3.

Creating the authorization profile for the Tivoli Workload Scheduler user

The two ways to create the authorization profile for the Tivoli Workload Scheduler user.

There are two alternative ways to perform this task:

- Using transaction `su02` and manually creating the profile.
- Using the Profile Generator (transaction `PFCG`).

Using transaction su02 and manually creating the profile

Perform the following steps:

1. Write a profile name, for example Z_TWS, and a description.
2. Manually add the authorizations according to the following table:

Object	Description	Authorization
S_ADMI_FCD	System authorizations	S_ADMI_ALL
S_APPL_LOG	Application logs	S_APPL_L_E2E
S_BTCH_ADM	Background processing: Background administrator	S_BTCH_ADM
S_BTCH_JOB	Background processing: Operations on background jobs	S_BTCH_ALL
S_BTCH_NAM	Background processing: Background user name	S_BTCH_ALL
S_DEVELOP	ABAP Workbench: full authorization to modify objects of type PROG	E_ABAP_ALL
S_LOG_COM	Authorization to run external commands	S_LOGCOM_ALL
S_PROGRAM	ABAP: program run checks	S_ABAP_ALL
S_RFC	Authoritation. check for RFC access	S_RFC_ALL
S_RZL_ADM	CCMS: System Administration	S_RZL_ALL
S_SPO_ACT	Spool: Actions	S_SPO_ALL
S_SPO_DEV	Spool: Device authorizations	S_SPO_DEV_AL
S_XMI_LOG	Internal access authorizations for XMI log	S_XMILOG_ADM
S_XMI_PROD	Authorization for external management interfaces (XMI)	S_XMI_ADMIN

The authorizations are located in the "Basis: Administration" object class.

Depending on the version of SAP R/3, the authorization S_RFC_ALL are located either in the "Cross-application Authorization Objects" or in the "Non-application-specific Authorization Objects" object class.

3. Save the profile.
4. Go to the user maintenance panel and assign the profile to the Tivoli Workload Scheduler SAP R/3 user.
5. Save the user data.

Using transaction PFCG (Profile Generator)

Perform the following steps:

1. Write a name, for example ZTWS, in **Role Name**.
2. Click **Create Role** and write a description for the role, such as "Role for the TWS user."
3. Save the role.
4. Select **Authorizations**.
5. Click **Change Authorization Data**.
6. In the pop-up, select **Templates**.
7. Manually add the following authorization objects:

Object	Description
S_ADMI_FCD	System authorizations
S_APPL_LOG	Application logs
S_BTCH_ADM	Background processing: Background administrator
S_BTCH_JOB	Background processing: Operations on background jobs
S_BTCH_NAM	Background processing: Background user name
S_PROGRAM	ABAP: Program run checks
S_DEVELOP	ABAP Workbench: full authorization to modify objects of type PROG
S_LOG_COM	Authorization to run external commands
S_RFC	Authorization check for RFC access
S_RZL_ADM	CCMS: System Administration
S_SPO_ACT	Spool: Actions
S_SPO_DEV	Spool: Device authorizations
S_XMI_LOG	Internal access authorizations for XMI log
S_XMI_PROD	Authorization for external management interfaces (XMI)

8. Fill in the values according to the following scheme:

Object	Description
S_ADMI_FCD	System authorizations • System administration function: Full authorization
S_APPL_LOG	Activity: Display • Application log Object name: Full authorization • Application log subobject: Full authorization
S_BTCH_ADM	Background processing: Background administrator • Background administrator ID: Full authorization
S_BTCH_JOB	Background processing: Operations on background jobs • Job operations: Full authorization • Summary of jobs for a group: Full authorization
S_BTCH_NAM	Background processing: Background user name • Background user name for authorization check: Full authorization
S_PROGRAM	ABAP: Program run checks • User action ABAP/4 program: Full authorization • Authorization group ABAP/4 program: Full authorization
S_RFC	Authorization check for RFC access • Activity: Full authorization • Name of RFC to be protected: Full authorization • Type of RFC object to be protected: Full authorization
S_RZL_ADM	Activity: Full authorization

Object	Description
S_SPO_ACT	Spool: Actions • Authorization field for spool actions: Full authorization • Value for authorization check: Full authorization
S_SPO_DEV	Spool: Device authorizations • Spool - Long device names: Full authorization
S_XMI_LOG	Internal access authorizations for XMI log • Access method for XMI log: Full authorization
S_XMI_PROD	Authorization for external management interfaces (XMI) • XMI logging - Company name: TIVOLI* • XMI logging - Program name: MAESTRO* • Interface ID: Full authorization

9. Save the authorizations.
10. Generate a profile. Use the same name that you wrote in **Role Name**.
11. Exit the authorization management panel and select **User**.
12. Add the Tivoli Workload Scheduler user to the role.
13. Save the role.

Copying the correction and transport files

The setup file loads four correction and transport files into the Tivoli Workload Scheduler home directory. Copy these correction and transport files to the SAP R/3 server and import them into the SAP R/3 database, as follows:

1. On your SAP R/3 database server, log on to the SAP R/3 system as an administrator.
2. Copy the control file and data file from the `TWS_home\methods` directory to the following directories on your SAP R/3 database server:

   ```
   copy control_file /usr/sap/trans/cofiles/
   copy data_file /usr/sap/trans/data/
   ```

   The names of *`control_file`* and *`data_file`* vary from release to release. The files are located in *`TWS_home`*`\methods` and have the following file names and format:

   For SAP R/3 releases earlier than 6.10:
   - `K000`*`xxx`*`.TV1` (control file) and `R000`*`xxx`*`.TV1` (data file)
   - `K900`*`xxx`*`.TV2` (control file) and `R900`*`xxx`*`.TV2` (data file)

   For SAP R/3 releases 6.10, or later:
   - `K9000`*`xx`*`.TV1` (control file) and `R9000`*`xx`*`.TV1` (data file)
   - `K9007`*`xx`*`.TV1` (control file) and `R9007`*`xx`*`.TV1` (data file)

   where *x* is a digit generated by the SAPsystem.

Specifically, for Tivoli Workload Scheduler for Applications version 8.5 the following files are used:

For SAP R/3 releases earlier than 6.10:
- `K000538.TV1` (for standard jobs scheduling)
- `R000538.TV1` (for standard jobs scheduling)
- `K900294.TV2` (for IDoc monitoring and job throttling)

- R900294.TV2 (for IDoc monitoring and job throttling)

For SAP R/3 releases 6.10, or later:

- K900044.TV1 (for standard jobs scheduling)
- R900044.TV1 (for standard jobs scheduling)
- K900751.TV1 (for IDoc monitoring and job throttling)
- R900751.TV1 (for IDoc monitoring and job throttling)

Importing ABAP/4 function modules into SAP R/3

How to generate, activate and commit new ABAP/4 modules to a SAP R/3 system.

This section describes the procedure to generate, activate, and commit new ABAP/4 function modules to your SAP R/3 system and several new internal tables. You do not modify any existing SAP R/3 system objects.

The number of ABAP/4 modules that you install with the import process varies from release to release. The modules are installed in the *TWS_home*\methods directory and have the following file names and format:

For SAP R/3 releases earlier than 6.10:

- K000*xxx*.TV1 (function modules for standard jobs scheduling extensions)
- K900*xxx*.TV2 (function modules for IDoc monitoring and job throttling)

For SAP R/3 releases 6.10, or later:

- K9000*xx*.TV1 (function modules for standard jobs scheduling extensions)
- K9007*xx*.TV1 (function modules for IDoc monitoring and job throttling)

where *x* is a digit generated by the SAP system.

To import ABAP/4 function modules into SAP R/3:

1. Change to the following directory:
   ```
   cd /usr/sap/trans/bin
   ```
2. Add the transport file to the buffer:
   ```
   tp addtobuffer transport sid
   ```
   where:

   transport
   The transport request file.

   sid The SAP R/3 system ID.

   For example, if the transport file in the *TWS_home*\methods directory is named K900*xxx*.TV1, the transport request is tv1K900*xxx*.
3. Run the **tp tst** command to test the import:
   ```
   tp tst transport sid
   ```
   After running this command, examine the log files in the /user/sap/trans/log directory for error messages. Warnings of severity level 4 are normal.

   If there are errors, check with a person experienced in correction and transport, or try using unconditional modes to do the import.
4. Run the following command to import all the files in the buffer:
   ```
   tp import transport sid
   ```
   This command generates the new ABAP/4 modules and commits them to the SAP R/3 database. They automatically become active.

After running this command, examine the log files located in the /user/sap/trans/log directory for error messages. Warnings of severity level 4 are normal.

If a problem is encountered, use unconditional mode when running this step:

```
tp import transport sid U126
```

5. When the import is complete, check the log files located in the /usr/sap/trans/log directory to verify that the ABAP/4 modules were imported successfully.

   The number of ABAP/4 modules installed by the import process varies depending on the SAP release you are using. If you are using an SAP release earlier than 6.10, 48 modules are installed. If you are using SAP release 6.10 or later and you apply the standard transport and the IDOC transport, 26 modules are installed. For a list of the transport files to be used, refer to "Importing ABAP/4 function modules into SAP R/3" on page 84. Table 31 lists the ABAP modules installed according to the SAP release.

Table 31. ABAP/4 modules installed according to the SAP release

ABAP/4 module	SAP releases earlier than 6.10	SAP releases 6.10 or later
ENQUEUE_/IBMTWS/EQ_XAPPL	✓	✓
DEQUEUE_/IBMTWS/EQ_XAPPL	✓	✓
/IBMTWS/UNREGISTER_XAPPL	✓	✓
/IBMTWS/GET_XAPPL_REGISTRATION	✓	✓
/IBMTWS/MODIFY_JOB_CLASS	✓	✓
/IBMTWS/REGISTER_XAPPL	✓	✓
J_1O1_BDC_STATUS	✓	✓
J_1O1_DATE_TIME	✓	✓
J_1O1_IDOC_SELECT	✓	✓
J_1O1_JOB_ADJUST_CLIENT	✓	✓
J_1O1_JOB_ABAP_STEP_MODIFY	✓	
J_1O1_JOB_ADD_ABAP_STEP	✓	
J_1O1_JOB_ADD_EXTPGM_STEP	✓	
J_1O1_JOB_CLOSE	✓	
J_1O1_JOB_COPY	✓	
J_1O1_JOB_DEFINITION_GET	✓	
J_1O1_JOB_DELETE	✓	
J_1O1_JOB_EXTPGM_STEP_MODFY	✓	
J_1O1_JOB_FIND	✓	✓
J_1O1_JOB_FINDALL	✓	✓
J_1O1_JOB_HAS_EXTENDED_VARIANT	✓	✓
J_1O1_JOB_LOG	✓	✓
J_1O1_JOB_LOG_READ	✓	
J_1O1_JOB_OPEN	✓	
J_1O1_JOB_START	✓	
J_1O1_JOB_STATUS	✓	
J_1O1_JOB_STATUS_CHECK	✓	

Table 31. ABAP/4 modules installed according to the SAP release (continued)

ABAP/4 module	SAP releases earlier than 6.10	SAP releases 6.10 or later
J_1O1_JOB_STOP	✓	
J_1O1_RAISE_EVENT	✓	✓
J_1O1_REPORT_ALL_SELECTIONS	✓	✓
J_1O1_REPORT_DYNSEL_GET_DEF	✓	
J_1O1_REPORT_GET_TEXTPOOL	✓	✓
J_1O1_TABLE_GET_NAMETAB	✓	
J_1O1_VARIANT_COPY	✓	✓
J_1O1_VARIANT_COPY_4X	✓	
J_1O1_VARIANT_CREATE	✓	✓
J_1O1_VARIANT_CREATE_NEW	✓	
J_1O1_VARIANT_DELETE	✓	✓
J_1O1_VARIANT_DELETE_NEW	✓	
J_1O1_VARIANT_EXISTS	✓	✓
J_1O1_VARIANT_GET_DEFINITION	✓	✓
J_1O1_VARIANT_GET_DEF_NEW	✓	
J_1O1_VARIANT_GET_HELP_VALUES	✓	✓
J_1O1_VARIANT_INFO_GET	✓	
J_1O1_VARIANT_MAINTAIN_CNT_TBL	✓	✓
J_1O1_VARIANT_MAINTAIN_SEL_TBL	✓	✓
J_1O1_VARIANT_MODIFY	✓	✓
J_1O1_VARIANT_MODIFY_NEW	✓	

Table 32 shows the contents of the ABAP modules for the IDoc records and job throttling feature.

Table 32. ABAP/4 modules contents

Object	Description	Used by...
/IBMTWS/	Type = Development Namespace. For IBM Tivoli Workload Scheduler.	Internal use only
/IBMTWS/EQ_XAPPL	Type = Lock Object. Synchronizes the job throttler instances and job interception collector jobs that are running against the same SAP system.	Job throttling Job interception
/IBMTWS/GET_ XAPPL_REGISTRATION	Type = Function Module. It is used to query for existing external application registration data in table IBMTWS/XAPPL, for example the registration data of a job throttler instance or job interception collector.	Job throttling Job interception
/IBMTWS/ MODIFY_JOB_CLASS	Type = Function Module. Modifies the job class of an intercepted job that is controlled by the job throttler. For details, see “Step 3. Enabling job class inheritance” on page 175.	Job throttling Job interception

Table 32. ABAP/4 modules contents (continued)

Object	Description	Used by...
/IBMTWS/ REGISTER_XAPPL	Type = Function Module. Registers an external application, for example the job throttler.	Job throttling Job interception
/IBMTWS/TWS4APPS	Type = Function group. For IBM Tivoli Workload Scheduler.	Internal use only
/IBMTWS/ UNREGISTER_XAPPL	Type = Function Module. Unregisters an external application, for example the job throttler.	Job throttling Job interception
/IBMTWS/XAPPL	Type = Table. Stores the registration data of external applications. An external application can be a job throttler instance or a job interception collector.	Job throttling Job interception
J_1O1_IDOC_SELECT	Type = Function Module. Selects IDoc records from SAP internal tables. For details, see "Defining event rules based on IDoc records" on page 189.	IDoc event rules
J_1O1_TWS_EDIDC	Type = Data structure in FM interface	Function module J_1O1_IDOC_SELECT
J_1O1_TWS_IDOC_ SELECTION	Type = Data structure in FM interface	Function module J_1O1_IDOC_SELECT
J_1O1_TWS_STATE_ SELECTION	Type = Data structure in FM interface	Function module J_1O1_IDOC_SELECT

Changing the Tivoli Workload Scheduler RFC user ID password

If the password of the Tivoli Workload Scheduler RFC user ID is modified after the initial installation, the options file used by `r3batch` must be updated with this change.

In UNIX, log on as root to the system where Tivoli Workload Scheduler is installed.

In Windows, log on as an administrator and start a DOS shell on the system where Tivoli Workload Scheduler is installed, as follows:

1. Generate an encrypted version of the new password using the **enigma** command in *TWS_home*/`methods`. To do this in a command shell, type:

   `enigma` *newpwd*

   where *newpwd* is the new password for the Tivoli Workload Scheduler RFC user ID.

   The **enigma** command prints an encrypted version of the password.
2. Copy the encrypted password into the options file, which is located in the *TWS_home*/`methods` directory. The file can be edited with any text editor.

Ensure that you copy the password exactly, preserving uppercase, lowercase, and punctuation. The encrypted password looks similar to:

```
{3des}Hchwu6IsF5o=
```

If the encrypted password is not entered correctly, Tivoli Workload Scheduler is unable to start or monitor SAP R/3 batch jobs.

Securing data communication

You can increase the security of your SAP system through the use of an external security product. Secure Network Communications (SNC) can integrate the external security product with the SAP system.

Data communication paths between the client and server components of the SAP system that use the SAP protocols RFC or DIAG are more secure with SNC. The security is strengthened through the use of additional security functions provided by an external product that are otherwise not available with SAP systems.

SNC provides security at application level and also end-to-end security. IBM Tivoli Workload Scheduler for Applications, Version 8.6 is extended to read SNC configuration parameters and forward them to the SAP RFC communication layer used when logging in to the SAP system. IBM Tivoli Workload Scheduler for Applications does not provide or ship SNC software but instead enables the use of third-party SNC products to secure the RFC communication.

Levels of protection

You can apply one of the following levels of protection:

Authentication only
This is the minimum level of security protection available with SNC. The system verifies the identity of the communication partners.

Integrity protection
The system detects any changes to the data which might have occurred between the two components communicating.

Privacy protection
This is the maximum level of security protection available with SNC. The system encrypts the messages being transferred so that any attempt to eavesdrop is useless. Privacy protection also includes integrity protection of the data.

The following options in the local options file are used to configure SNC for Tivoli Workload Scheduler for Applications:

- **r3snclib**: the path and file name of the SNC library.
- **r3sncmode**: enables or disables SNC between r3batch and the SAP R3 system.
- **r3sncmyname**: the name of the user sending the RFC for SNC.
- **r3sncpartnername**: the SNC name of the SAP R3 communication partner (application server).
- **r3sncqop**: the SNC protection level.

See "Defining the local options" on page 92 for a description of these options in the local options file.

Migrating from previous versions

This version of the Tivoli Workload Scheduler agent workstation for SAP R/3 supports all the SAP R/3 versions listed in Table 21 on page 71. The Tivoli Workload Scheduler agent workstation for SAP R/3 Version 8.6 uses the official SAP R/3 RFC interfaces for job scheduling. These are:

- The BC-XBP 6.10 (V2.0) Interface function modules for SAP R/3 versions 6.10 and later.
- The BC-XBP 7.00 (V3.0) Interface function modules for SAP R/3 versions 7.00, Support Package 16, in addition to the BC-XBP 6.10 (V2.0) function modules.

To avoid conflicts with other vendors, the Tivoli Workload Scheduler ABAP modules now belong to the Tivoli Workload Scheduler partner namespace J_1O1_*xxx*and /IBMTWS. After you have completed the imports as described in "Importing ABAP/4 function modules into SAP R/3" on page 84, the RFC J_1O1_*xxx* function modules and the /IBMTWS function modules are installed on your system.

If you had a previous installation of Tivoli Workload Scheduler extended agent for SAP R/3 on your system, you can delete the following function modules from your SAP R/3 system:

```
Z_MAE2_BDC_STATUS
Z_MAE2_DATE_TIME
Z_MAE2_JOB_COPY
Z_MAE2_JOB_DELETE
Z_MAE2_JOB_FIND
Z_MAE2_JOB_FINDALL
Z_MAE2_JOB_LOG
Z_MAE2_JOB_OPEN
Z_MAE2_JOB_START
Z_MAE2_JOB_STATUS
Z_MAE2_JOB_STOP
```

These are old versions of the Tivoli ABAP functions, which belong to the customer name space. You can also delete the function group YMA3. It is not necessary to delete the function modules and the function group, but delete them if you want to clean up your system.

Print parameter and job class issues

The workstation running the r3batch access method for SAP R/3 uses the official RFC interfaces of SAP R/3 for job scheduling. When you migrate from previous versions of SAP R/3, there can be problems with print parameters in jobs launched by Tivoli Workload Scheduler. This is because of limitations in the RFC interfaces.

These limitations are no longer true with XBP 2.0 and later.

The following is a list of print parameters supported by BAPI XBP 1.0 for SAP R/3 release 4.6*x* and later:
- archiving mode
- authorization
- columns
- delete after output
- lines
- number of copies
- output device
- print immediately
- recipient
- sap cover page
- selection cover page
- spool retention period

To resolve the loss of print parameters when copying a job, install the appropriate SAP R/3 Support Package as stated in the SAP R/3 notes 399449 and 430087.

The same applies to the job class. Official SAP R/3 interfaces only allow class C jobs. Installing the SAP R/3 Support Package also resolves this issue.

Configuring the SAP R/3 access method

This section provides detailed information about the SAP options files creation. The files for the SAP access method are located in `TWS_home/methods`.

Extended agents

r3batch.opts

A common configuration file for the `r3batch` access method, whose settings affect all the `r3batch` instances. It functions as a “global” configuration file.

***XANAME*_r3batch.opts**

A configuration file that is specific to each Tivoli Workload Scheduler extended agent workstation that uses the `r3batch` access method. Its options affect only the `r3batch` instance that is used by that particular workstation. It functions as a “local” configuration file.

Note: `XANAME` is the name of the extended agent workstation and must be written in uppercase characters.

For example, to define two extended agents named `wkst1` and `wkst2` that access two SAP systems, `SAP1` and `SAP2`, with the `r3batch` access method, you must define the following three configuration files:

- Global `r3batch.opts`
- Local file `WKST1_r3batch.opts`
- Local file `WKST2_r3batch.opts`

Dynamic agents

r3batch.opts

A common configuration file for the `r3batch` access method, whose settings affect all the `r3batch` instances. It functions as a “global” configuration file.

DYNAMIC_AGENT_FILE_r3batch.opts

One or more configuration files that are specific to each dynamic agent workstation within a particular installation of a `r3batch` access method. The `DYNAMIC_AGENT_r3batch.opts` is the name of the option file, where *DYNAMIC_AGENT* is not necessarily the name of the dynamic agent workstation, because the dynamic agent can have more than one`.opts` file associated. If you do not create a local options file, the global options file is used. Every dynamic agent workstation must have one or more local options file with its own configuration options.

Note: The value for *DYNAMIC_AGENT* must be written in uppercase alphanumeric characters. Double-byte character set (DBCS), Single-byte character set (SBCS), and Bidirectional text are not supported.

Note: If you have a **pool** or **dynamic pool** containing *n* agents, you have to create an option file for the dynamic pool and copy it in the `TWA_DIR/TWS/methods` of each agent of the pool. Then you have to create in the same directory another option file for the

specific agent.
For example, if the SAP R/3 access method is installed for **AGENT1** and **AGENT2** that belong to the dynamic pool ***DYN_POOL***, you need to create in the TWA_DIR/TWS/methods of each agent the following option files:

AGENT 1

- FILE_AGENT1_r3batch.opts
- DYN_POOL_r3batch.opts

AGENT2

- FILE_AGENT2_r3batch.opts
- DYN_POOL_r3batch.opts

.

Note:

1. To successfully use the SAP R/3 access method, you must first install the SAP RFC libraries, as described in “Software requirements” on page 71.
2. If r3batch finds the local configuration file for an extended agent or dynamic agent, it ignores the duplicate information contained in r3batch.opts. If instead it does not find a local configuration file then it will use r3batch.opts.

Defining the configuration options

This section describes the options you can configure in r3batch.opts and in *XANAME*_r3batch.opts.

Defining the global options

Table 33 lists the options that can be specified only in the *global* configuration file r3batch.opts.

Table 33. r3batch global configuration options

Option	Description	Default
dep_sem_proj	(Optional) The project ID for the external dependency semaphore used for handling SAP R/3 background processing events as external follows dependencies.	d
icp_sem_proj	(Optional) The project ID for the job interception semaphore.	c
job_sem_proj	(Optional) The project ID for the job semaphore.	a
primm_enable	(Optional) Enables (ON) the SAP print parameter PRIMM (Print Immediately) for all jobs.	OFF
prnew_enable	(Optional) Enables (ON) the SAP print parameter PRNEW (New Spool Request) for all jobs.	OFF
prrel_enable	(Optional) Enables (ON) the SAP print parameter PRREL (Immediately delete the spool output after printing) for all jobs.	OFF
prsap_enable	(Optional) Enables (ON) the SAP print parameter PRSAP (Print SAP Cover Page) for all jobs. The default value is OFF.	OFF
prunx_enable	(Optional) Enables (ON) the SAP print parameter PRUNX (Print Operating System Cover Page) for all jobs.	OFF
var_sem_proj	(Optional) The project ID for the variant semaphore.	b

Modifying the default values of the semaphore options is particularly useful when the IDs that are generated would be the same as the IDs already used by other applications.

On UNIX and Linux, to resolve the problem of duplicated IDs, Tivoli Workload Scheduler for Applications for SAP uses system-5 semaphores to synchronize critical ABAP function module calls. It uses one semaphore for job-related tasks and another one for tasks related to variant maintenance.

To synchronize on the same semaphore, the communication partners must use the same identifier. There are several ways to choose this identifier. Tivoli Workload Scheduler for Applications for SAP uses two parameters: a path name and a project ID (which is a character value). The path name parameter is the fully qualified path to the options file. The project ID is taken from the options described in Table 33 on page 91. If these options are omitted, Tivoli Workload Scheduler for Applications for SAP uses default values, which work for most installations.

Note:

1. The semaphore options must be edited directly in the global options file using a text editor; you cannot use the options editor to modify these values.
2. If two semaphore options are assigned the same value, all the semaphore values are reset according to the following rule:

   job_sem_proj
   It keeps the value assigned, or its default value.

   var_sem_proj
   It is reset to the first character that, in the ASCII table, follows the value assigned to `var_sem_proj`.

   icp_sem_proj
   It is reset to the second character that, in the ASCII table, follows the value assigned to `var_sem_proj`.

   dep_sem_proj
   It is reset to the third character that, in the ASCII table, follows the value assigned to `var_sem_proj`.

Defining the local options

Table 34 on page 93 lists the options that you can specify only in the *local* configuration files.

Table 34. r3batch local configuration options

Option	Description
bapi_sync_level	(Optional) Specifies the synchronization level between the SAP function modules BAPI_XBP_JOB_COPY and BAPI_XBP_JOB_START_ASAP. Allowed values are: **high** All RFC calls between BAPI_XBP_JOB_START_ASAP and BAPI_XBP_JOB_COPY are synchronized. This is the default. **medium** The RFC calls to BAPI_XBP_JOB_START_ASAP are synchronized. **low** The RFC calls are not synchronized.
blank_libpath	(Optional) Clears (ON) the operating system variables LD_LIBRARY_PATH and LIBPATH. The default value is OFF.
get_job_status_retry	(Optional) Sets the number of times a Remote Function Call must be attempted to retrieve the actual status of an SAP Job. Allowed values are in the range from 1 to 9999. The default value is 5.
get_job_status_retry_delay	(Optional) Sets the number of seconds between two consecutive calls of a Remote Function Call. Allowed values are in the range from 1 to 9999.
job_duration	(Optional) Enables (ON) that the CPU time value in the production plan report that is run from the Dynamic Workload Console is set to the actual duration of the SAP job. Default value is OFF. To retrieve the job duration from the SAP system, ensure that the authorization profile contains the following authorization objects: • S_DEVELOP • S_TCODE with parameter SE38 (only for SAP 6.40 and 7.00) For details about the authorization profile, see "Creating the authorization profile for the Tivoli Workload Scheduler user" on page 80.
primm_enable	(Optional) Enables (ON) the SAP print parameter PRIMM (Print Immediately) for all jobs. The default value is OFF.
prnew_enable	(Optional) Enables (ON) the SAP print parameter PRNEW (New Spool Request) for all jobs. The default value is OFF.
prrel_enable	(Optional) Enables (ON) the SAP print parameter PRREL (Print Release) for all jobs. The default value is OFF.
prsap_enable	(Optional) Enables (ON) the SAP print parameter PRSAP (Print SAP Cover Page) for all jobs. The default value is OFF.
prunx_enable	(Optional) Enables (ON) the SAP print parameter PRUNX (Print Operating System Cover Page) for all jobs. The default value is OFF.
r3client	(Mandatory) The SAP client number.
r3gateway	(Optional) The host name of the SAP gateway.
r3group	(Optional) The name of the SAP logon group.
r3gwservice	(Optional) The service number of the SAP gateway.

Table 34. r3batch local configuration options (continued)

Option	Description
r3host	(Mandatory) The host name of the SAP message server when using logon groups, or the host name of the application server in all other cases. If this server can be reached through one or more SAP gateways, use a string in the format `/H/gateway/H/` for each of them.
r3instance	(Mandatory) The SAP instance number. If **r3group** is set, this option is ignored.
r3password	(Mandatory) The password for the **r3user**. Ensure that you enter the same password when creating this user in the SAP system. It can be a maximum of eight characters and is stored in encrypted format. The value is case sensitive. For information about how to encrypt the password see "Encrypting SAP R/3 user passwords" on page 102.
r3sid	(Mandatory) The SAP system ID.
r3snclib	(Optional) Specifies the path and file name of the SNC library. This option becomes mandatory if **r3sncmode** is activated (1).
r3sncmode	(Optional) Enables (1), or disables (0), secure network communication (SNC) between r3batch and the SAP R3 system. The default setting is (0). Refer to the SAP documentation for more information about using the SAP cryptographic Library for SNC.
r3sncmyname	(Optional) Specifies the name of the user sending the RFC for secure network communication (SNC).
r3sncpartnername	(Optional) Specifies the SNC name of the SAP R3 communication partner (application server). This option becomes mandatory if **r3sncmode** is activated (1).
r3sncqop	(Optional) Specifies the secure network communication (SNC) protection level.
r3user	(Mandatory) The name of the SAP user with which the access method connects to the SAP system. It must have the appropriate privileges for running background jobs. It is sometimes also called the Maestro User ID.
report_list_max_limit	(Optional) Sets the maximum number of ABAP reports which can be loaded. The default value is -1, which means no limit.

Defining the common options

Table 35 on page 95 lists additional options that you can specify in either configuration file.

Table 35. r3batch common configuration options

Option	Description	Default
bdc_job_status_failed	(Optional) How Tivoli Workload Scheduler sets the completion status of a job running BDC sessions, according to a possible BDC processing failure. The allowed values are: *n* If at least *n* BDC sessions failed (where *n* is an integer greater than 0), Tivoli Workload Scheduler sets the job completion status as failed. **all** If all the BDC sessions failed, Tivoli Workload Scheduler sets the job completion status as failed. **ignore** When all the BDC sessions complete, regardless of their status, Tivoli Workload Scheduler sets the job completion status as successful. This is the default. **Note:** This option is ignored if you defined the job by setting the **nobdc** or **nobdcwait** option. For details about these options, see "Task string to define SAP jobs" on page 115.	ignore
ccms_alert_history	(Optional) Enables (ON) or disables (OFF) the product to retrieve all the matching CCMS alerts, included those that were generated before the monitoring process started. The default value is OFF, meaning that the product retrieves only the CCMS alerts that are generated after the monitoring process started. **Note:** This option takes effect the first time you start the CCMS alert monitoring. If you initially set it to OFF and later you want to retrieve the alerts generated before the monitoring process started, stop the monitoring and delete the *XAname*_r3xalmon.cfg file located in *TWS_home*/methods/r3evmon_cfg. In the options file, set ccms_alert_history=on and start the monitoring process again.	OFF
commit_dependency	(Optional) Enables (ON) or disables (OFF) the product to commit internetwork dependencies after processing. If you enable this option, internetwork dependencies are committed immediately by default. If you disable or delete this option, the **-commit** parameter set in the internetwork dependency definition is applied. For details about the **-commit** parameter, see Table 45 on page 183.	OFF
enable_appl_rc	(Optional) Enables (ON) or disables (OFF) the mapping of the application return code to the Tivoli Workload Scheduler return code. **Note:** This feature does not modify the exit code of the access method. For more details, refer to the **rccondsucc** keyword in the job definition documented in *Tivoli Workload Scheduler: User's Guide and Reference*.	OFF
evmon_interval	(Optional) The polling rate (in seconds) that the r3evmon process applies to monitor the list of events.	60
ifuser	(Optional) The ID of the user who runs the access method to retrieve job information.	None
idoc_no_history	(Optional) Enables (ON) or disables (OFF) the product to retrieve only IDoc data that is generated after the monitoring process started. If you specify OFF, all matching IDocs are retrieved, including those that were generated before the monitoring process started. When processing this option, r3evmon uses the *XAname*_r3idocmon.cfg file to retrieve the date and time for the next monitoring loop.	ON

Table 35. r3batch common configuration options (continued)

Option	Description	Default
`idoc_shallow_result`	(Optional) Enables (`ON`) or disables (`OFF`) the product to retrieve only the most recent matching IDocs. For example, suppose you set `idoc_shallow_result=ON`. If the status of an IDoc changes several times during the monitoring interval and the same status, matching an event rule condition, occurs more than once in the sequence of statuses, only the most recent matching IDoc is retrieved. If you specify `OFF`, all matching IDocs are retrieved.	`ON`
`jobdef`	(Optional) If enabled, you can use the Dynamic Workload Console to define jobs, in addition to the command line. Specify `r3batch` to enable the option, and any other value to disable it.	`r3batch`
`job_interceptable`	(Optional) Enables (`ON`) or disables (`OFF`) the job launched by r3batch to be intercepted by SAP. If enabled, when r3batch launches a job and the SAP job interception feature is enabled, the job can be intercepted if it matches previously defined criteria. If disabled, the job launched by r3batch cannot be intercepted by SAP.	`OFF`
`ljuser`	(Optional) The ID of the user who runs the access method to launch jobs (LJ tasks) and manage jobs (MJ tasks).	None
`log_r3syslog`	(Optional) Enables (`ON`) or disables (`OFF`) the access method to write the latest entries from the SAP syslog to its trace file when an RFC returns with a general error.	`OFF`
`long_interval`	(Optional) The maximum interval, in seconds, between status checks. It cannot be greater than 3600 seconds. See also *short_interval*.	`3600`
`max_n0_counter`	(Optional) The maximum value of the N0 counter. If the N0 counter reaches the specified value, it starts again from 0.	`2^15 - 1`
`max_name_counter`	(Optional) The maximum value of the variant name counter. If the name counter reaches the specified value, it starts again from 0.	`40`
`n0_counter_policy`	(Optional) The N0 counter policy: **step** The N0 counter is increased once for every step. **job** The N0 counter is increased once for every job.	`job`
`name_counter_policy`	(Optional) The name counter policy: **step** The name counter is increased once for every step. **job** The name counter is increased once for every job.	`job`
`nojobdefs`	(Optional) Disables (`1`) or enables (`0`) the definition of new SAP jobs using the Dynamic Workload Console. If this option is set to 1, you must create the job definitions in the SAP job before creating the Tivoli Workload Scheduler for Applications job that is going to schedule them.	`0`
`oldcopy`	(Optional) Enables (`1`) or disables (`0`) the access method to use the old way of copying jobs, even though the function module BAPI_XBP_JOB_COPY is present on the SAP system.	`0`

Table 35. r3batch common configuration options (continued)

Option	Description	Default
pchain_recover	(Optional) The action taken by Tivoli Workload Scheduler when you rerun a job that submits a process chain. The allowed values are: **rerun** Tivoli Workload Scheduler creates another process chain instance and submits it to be run again. **restart** Tivoli Workload Scheduler restarts the original process chain from the failing processes to the end. For details about rerunning a process chain job, refer to "Rerunning a process chain job" on page 167.	`rerun`
pchain_details	(Optional) Enables (`ON`) or disables (`OFF`) the display of details about an SAP process chain that you scheduled as a Tivoli Workload Scheduler job.	`OFF`
pchainlog_bapi_msg	(Optional) Enables (`ON`) or disables (`OFF`) the product to retrieve additional messages from the BAPI calls from the SAP Business Warehouse process chains and appends them to the stdlist of Tivoli Workload Scheduler.	`ON`
pchainlog_level	(Optional) Supplements the option **retrieve_pchainlog**. Specifies which level of process chain logs you want to retrieve. Allowed values are: **1** Only the first level of process chain is logged. *level_number* Process chains are logged down to the level of chain you indicate here. For example, if you indicate 2 only the first two levels are logged. **all** All process chains are logged.	If you omit this option, and leave **retrieve_pchainlog** set to `ON`, the default is level 1.
pchainlog_verbosity	(Optional) Supplements the option **retrieve_pchainlog**. Specifies which type of process chain logs you want to retrieve. Allowed values are: **chains_only** Logs only the process chains. **chains_and_failed_proc** In addition to the process chains, logs all failed processes. **complete** Logs all process chains and processes. **Note:** This option affects the entire process chain; verbosity cannot be reduced for individual processes.	If you omit this option, and leave **retrieve_pchainlog** set to `ON`, the default is `complete`.
pc_launch_child	(Optional) Enables (`ON`) or disables (`OFF`) the product to launch child jobs that are in scheduled state. **Note:** You can use this option only if you activated the parent-child feature on the SAP system. On the XBP 2.0 or later SAP system, you can activate this feature by using the INITXBP2 ABAP report.	`OFF`
placeholder_abap_step	(Optional) If XBP version 2.0 is used, the name of the ABAP report used as the dummy step in the SAP placeholder job that is created to monitor an SAP event defined as external dependency.	If this option is not specified, either as global or local option, the default `BTCTEST` is used.

Table 35. r3batch common configuration options (continued)

Option	Description	Default
qos_disable	(Optional) Enables (`ON`) or disables (`OFF`) the creation of the environment variable QOS_DISABLE on Microsoft Windows systems that use the Quality of Service (QoS) feature, before `r3batch` opens an RFC connection. Without this option, because of problems in the implementation of the QoS service, the connection between `r3batch` and the SAP RFC library does not work.	`OFF`
r3auditlevel	(Optional) The audit level for the XBP. A number from 0 (low) to 3 (high).	3
rcmap	(Optional) Enables (`ON`) or disables (`OFF`) the return code mapping capabilities of Tivoli Workload Scheduler for Applications for SAP.	`ON`
retrieve_applinfo	(Optional) Enables (`ON`) or disables (`OFF`) the retrieval and appending of the SAP application log to the stdlist of Tivoli Workload Scheduler.	`OFF`
retrieve_ipaklog	(Optional) Enables (`ON`) or disables (`OFF`) the retrieval and appending of the SAP BW InfoPackage logs to the `stdlist` of Tivoli Workload Scheduler. **Note:** The retrieval and appending of SAP BW InfoPackage job logs to the `stdlist` might be time-consuming for jobs that produce large logs.	`ON`
retrieve_joblog	(Optional) Enables (`ON`) or disables (`OFF`) the retrieval and appending of the SAP job logs to the `stdlist` of Tivoli Workload Scheduler. **Note:** 1. The retrieval and appending of job logs to the `stdlist` might be time-consuming for jobs that produce large logs. 2. If you disable the retrieval of the job logs, you also disable the return code mapping function for the log entries. 3. This option does not affect the `BDC Wait` feature.	`ON`
retrieve_pchainlog	(Optional) Enables (`ON`) or disables (`OFF`) the retrieval and appending of the SAP BW process chain logs to the `stdlist` of Tivoli Workload Scheduler. **Note:** 1. The retrieval and appending of SAP BW process chain logs to the `stdlist` might be time-consuming for jobs that produce large logs. 2. If you disable the retrieval of the SAP BW process chain logs, you also disable the return code mapping function for the log entries. 3. This option on its own retrieves the log of only the first level of a process chain. To retrieve more complete logs, use this option with the **pchainlog_level** and **pchainlog_verbosity** options.	`ON`

Table 35. r3batch common configuration options (continued)

Option	Description	Default
retrieve_spoollist	(Optional) Enables (ON) or disables (OFF) the retrieval and appending of the SAP job spool lists to the stdlist of Tivoli Workload Scheduler. **Note:** 1. The retrieval and appending of SAP job spool lists to the stdlist might be time-consuming for jobs that produce large spool lists. 2. If you disable the retrieval of the SAP job spool lists, you also disable the return code mapping function for the spool list entries.	ON
retry	(Optional) The retry count for SAP function module calls. Specify an integer greater than 0.	5
rfc_interval	(Optional) The polling rate (in milliseconds) with which r3batch listens for results of RFC requests. The rate cannot exceed 1,000 milliseconds. Consider that the lower the value of the rfc_interval option, the higher the frequency with which RFC request results are collected and, as a consequence, CPU consumption on the r3batch system is high.	10
rfc_open_delay	(Optional) The maximum number of seconds to wait between two consecutive calls before opening an RFC connection.	1800
rfc_open_retry	(Optional) The retry count for opening an RFC connection to the SAP system. Specify an integer greater than 0 to limit the number of retries, or -1 for an unlimited number of retries.	5
rfc_timeout	(Optional) The time (in seconds) that r3batch waits before canceling a non-responding RFC communication. Allowed values are in the range from 0 to 9999; 0 means no timeout.	600
short_interval	(Optional) The minimum interval, in seconds, between status checks. It cannot be less than 2 seconds. Setting this option to low values makes the notification of status changes faster, but increases the load on the hosting machine. See also *long_interval*.	10
throttling_enable_ job_class_inheritance	(Optional) Enables (ON) or disables (OFF) the inheritance of priority class. ON means that the intercepted job inherits the priority class of its progenitor job, if it is higher than its own class; otherwise it keeps its own class. OFF means that the intercepted job keeps its own class, regardless of its progenitor's class. **Note:** By setting this option, the parent-child feature is automatically enabled on the SAP system.	ON
throttling_enable_ job_interception	(Optional) Enables (ON) the job interception feature at job throttler startup, or keeps the current setting (OFF). ON means that when the job throttler starts, it enables the job interception feature on the SAP system. When the job throttler is stopped, the job interception feature is also automatically restored to the setting that was previously configured on the SAP system. OFF means that the job interception feature is kept as it is currently set in the SAP system.	ON

Table 35. r3batch common configuration options (continued)

Option	Description	Default
throttling_job_ interception_ version	Specifies the BC-XBP interface version to be used when the job throttler starts. Valid values are: • 2 • 3 The default BC-XBP interface version that is used is 2 (version 2.0).	2
throttling_interval	(Optional) The interval (in seconds) between each job throttling run.	5
throttling_max_ connections	(Optional) The maximum number of connections (connection pool size) that the job throttler can open to communicate with the SAP system. The minimum value is 3.	5
throttling_release_all_ on_exit	(Optional) Enables (ON) or disables (OFF) the release of all intercepted jobs. ON means that when the job throttler is stopped, it releases all the intercepted jobs. OFF means that when the job throttler is stopped, it does not release the intercepted jobs therefore the jobs remain intercepted, in scheduled state.	ON
throttling_send_ ccms_data	(Optional) Enables (ON) or disables (OFF) the sending of data from job throttling to the SAP CCMS Monitoring Architecture. ON means that the job throttler sends its status data to CCMS continuously. OFF means that the job throttler does not send its status to CCMS.	OFF
throttling_send_ ccms_rate	(Optional) Rate (in number of runs) at which the job throttler sends its status data to the SAP CCMS monitoring architecture. The minimum value is 1, meaning that the job throttler sends the data at every run.	1
twsmeth_cp	(Optional) The code page that r3batch uses to write its output. This option must be consistent with **twsmeth_lang**. It can be any of the existing TIS codepages.	The code page used by the Tivoli Workload Scheduler workstation that hosts the r3batch access method.
twsmeth_lang	(Optional) The language used to report messages. This option must be consistent with **twsmeth_cp**.	The language of the locale of the workstation that hosts the r3batch access method.
twsxa_cp	(Optional) The encoding used by r3batch to establish RFC connections with SAP systems. Use this option if r3batch is not Unicode-enabled. Possible values are: • 1100 • 1103 • 8000 • 8300 • 8400	1100

Table 35. r3batch common configuration options (continued)

Option	Description	Default
`twsxa_lang`	(Optional) The language used to log in to SAP systems. Specify one of the following (DE, EN, and JA can be set from the Option Editor. The other languages can be set using any text editor): **DE** German **EN** English **ES** Spanish **FR** French **IT** Italian **JA** Japanese **KO** Korean **pt_BR** Brazilian Portuguese **zh_CN** Simplified Chinese **zh_TW** Traditional Chinese **Note:** If you are working with InfoPackages and process chains on operating systems that do not support Unicode, this option must be set.	`EN`
`use_fips`	(Optional) Enables (ON) or disables (OFF) the FIPS mode of operation for Tivoli Workload Scheduler for Applications.	`OFF`
`utf8cmdline`	(Optional) Enables (`1`) or disables (`0`) the encoding of extended parameters in UTF-8 format. The default value is `0`.	**Note:** If you have both global and local options files and you want to change the default value for **utf8cmdline**, modify the local options file because this overrides the global options.
`variant_delay`	(Optional) The time, in seconds, that r3batch allows the SAP system to clean up the structures used for communication between r3batch and the SAP system. This option is valid when you launch a job that uses extended variants and requires a copy of a job template. Use this option only when you want to reduce r3batch response time, because it increases the load on the hosting machine. Higher values of **variant_delay** increase the response time and decrease the load. Allowed values are in the range from 0 to 3600.	`10`
`variant_selection_screen`	(Optional) Specifies the functional interface used to read report selection screens. Specify one of the following: **Custom** To communicate with the SAP system using the IBM Tivoli Workload Scheduler for Applications custom function module. **SAP** To communicate with the SAP system using the XBP 3.0 function module.	`Custom`
`xbpversion`	(Optional) The XBP version used on the target SAP system. Specify an integer value. This value overwrites the XBP version automatically determined during RFC logon. **Note:** For details about XBP 3.0 and SAP NetWeaver 2004s with SP9, refer to the SAP Note 977462.	The XBP version determined by r3batch during RFC logon from the SAP system.

SAP R/3 option file example

Below is an example of an options file for SAP. It can help you determine your specific site requirements, although your options file might be different.

```
r3client=100
r3host=/H/amsaix64.tivlab.raleigh.ibm.com
r3instance=00
r3password={3des}Hchwu6IsF5o=
r3sid=GS7
r3user=twstest
long_interval=120
r3auditlevel=3
short_interval=10
twsxa_lang=EN
```

Encrypting SAP R/3 user passwords

When you add your entries in the options file, the password value is automatically encrypted before it is written in the file. If you modify the file with a text editor, you must run the **enigma** program to encrypt the password before writing it in the file. To run the encryption program, enter the following command:

```
enigma [password]
```

You can include the password on the command line or enter it in response to a prompt. The program returns an encrypted version that you can then enter in the options file.

Starting from Tivoli Workload Scheduler for Applications version 8.4, a new algorithm is applied to encrypt the password that is stored in the options file. Use enigma, pwdcrypt, or the Option Editor to encrypt your password by applying this new algorithm. It is also applied if you open and save the options file with the Option Editor.

Configuration options usage

The format of the *XANAME*_r3batch.opts configuration file is the following:

```
option1=value1
option2=value2
option3=value3
...
```

with no blanks before the option, after the value, or before or after the equals (=) character.

You can put all the common information, such as the LJuser, IFuser, JobDef, and LogFileName options in r3batch.opts, while you can put tailored data for the target SAP system of the extended agent (for example, SAP1) in a local configuration file (for example, XA1_r3batch.opts).

You can put a local option in the global configuration file if you want to give the same option to all the r3batch instances. For example, if the SAP user name is the same in all your SAP systems, you can place the r3user option in the global file without duplicating that information in all the local configuration files.

A global option, such as job_sem_proj, only has effect in the global configuration file. If you put global options in a local file they have no effect.

r3batch reads the global configuration file first and then the local file. Every option (except the global options) contained in the local configuration file will override those in the global file. For example, if both the global and the local configuration files contain the `r3user` option, `r3batch` uses the one in the local file.

There are six mandatory options that `r3batch` requires:
- `r3client`
- `r3host`
- `r3instance`
- `r3password`
- `r3sid`
- `r3user`

You can put them all in the local configuration file or you can spread them between the global and the local files. For example, you could put `r3user` and `r3password` in the global configuration file and `r3sid`, `r3instance`, `r3client`, and `r3host` in the local one.

The `r3user` option is both local and mandatory. It must be put either in the global configuration file or the local configuration file.

Note: These configuration files are not created during the installation process.

Connecting to the SAP system

Tivoli Workload Scheduler for Applications for SAP uses the SAP remote connection call (RFC) library to connect to the SAP system. The connection address for an SAP system is denoted as a `connection string`.

Connecting to a specific application server

To connect to a specific application server, you enter strings which, according to the complexity of the networks, might be more or less complex and contain passwords to secure the routers.

In its basic form, a connection string consists of the host name (or IP name) of an SAP application server; for example:

```
/H/hemlock.tivlab.raleigh.ibm.com
```

This type of connection string works only in very simple network environments, where all application servers can be reached directly through TCP/IP. Usually, modern companies use more complex network topologies, with a number of small subnetworks, which cannot communicate directly through TCP/IP. To support this type of network, the SAP RFC library supports SAP routers, which are placed at the boundaries of the subnetworks and act as proxies. For this type of network, the connection string is a composite of basic connection strings for each SAP router, followed by the basic connection string for the target SAP system; for example:

```
 /H/litespeed/H/amsaix33/H/hemlock.tivlab.raleigh.ibm.com
```

Moreover, you can secure the SAP routers with passwords, to prevent unauthorized access. In this case, the basic connection string for the SAP router is followed by `/P/` and the password of the router.

Note: The SAP RFC library limits the length of the connection string to a maximum of 128 characters. This is a real limitation in complex network environments. As a workaround, it is recommended to use simple host names, without the domain name whenever possible. Alternatively, you can use the IP address, but this is not recommended, because it is difficult to maintain.

Tivoli Workload Scheduler for Applications for SAP supports both types of connection strings, basic and composite, where:

r3host The connection string.

r3instance
The SAP instance number.

r3sid The SAP system ID.

For example:

```
r3host=/H/litespeed/H/amsaix33/H/hemlock.tivlab.raleigh.ibm.com
r3instance=00
r3sid=TV1
```

Connecting to a logon group

In large SAP installations, the application servers are usually configured in logon groups for load balancing and fault-tolerance purposes. Load balancing is done by a dedicated server, called the `message server`. The message server automatically assigns users to the application server with the least workload of the logon group it controls.

Ensure that the file services (on UNIX: `/etc/services` on Windows: `C:\Windows\system32\drivers\etc\services`) contain an entry for the message server port of the SAP system to which `r3batch` connects. The entry has the following format:

```
sapmsSID 36system_number/tcp
```

where *SID* is the SAP system ID, and *system_number* is the SAP system number.

Set the following options to configure `r3batch` to connect to a logon group:

r3host The hostname of the message server.

r3group
The name of the logon group.

r3sid The SAP system ID.

For example:

```
r3host=pwdf0647.wdf.sap-ag.de
r3group=PUBLIC
r3sid=QB6
```

Configuring SAP event monitoring

This section provides detailed information about how to configure your system to monitor SAP events:

- “Prerequisite to defining event rules based on SAP events”
- “Monitoring SAP events” on page 105

Prerequisite to defining event rules based on SAP events

To be able to define event rules based on one or more SAP events, stop the Tivoli Workload Scheduler WebSphere Application Server and copy the following file (located on the system where you installed Tivoli Workload Scheduler for Applications,Version 8.6:

```
TWS_home/methods/SAPPlugin/SapMonitorPlugIn.jar
```

to the following directory of the master domain manager and of its backup nodes:

```
TWS_home/eventPlugIn
```

For the changes to take effect, stop and restart the Tivoli Workload Scheduler WebSphere Application Server. If the master domain manager is connected to the Dynamic Workload Console, stop and restart also the Dynamic Workload Console Application Server.

Monitoring SAP events

Whenever you define an event rule based on an SAP event in your Tivoli Workload Scheduler plan, that event is monitored by Tivoli Workload Scheduler. Monitoring SAP events is allowed only if you use XPB version 3.0, or later.

Tivoli Workload Scheduler monitors two types of SAP event:

Events defined by the SAP system
The events that are triggered automatically by system changes, for example when a new operation mode is activated. This type of event cannot be modified by the user.

Events defined by the user
The events that are triggered by ABAP or external processes, for example when a process triggers an SAP event to signal that external data has arrived and must be read by the SAP system. For details about how to trigger events by external processes, refer to “Raising an SAP event” on page 127.

If you modify the `r3batch` option files, to make the changes effective you must stop and restart the extended agent monitoring processes with the following command. For UNIX only, this command must be entered by the owner of the Tivoli Workload Scheduler installation:

Command syntax

```
►►──r3evman──┬─start─┬──────────────────────────────────────────────►◄
             └─stop──┘
```

Where:

start | stop
The action to perform:

start Starts monitoring SAP events.

stop Stops monitoring SAP events.

Defining SAP jobs

To define and manage jobs on an SAP workstation from Tivoli Workload Scheduler, you must define the following:

Jobs in SAP that you want to run under Tivoli Workload Scheduler control
You can define these jobs using standard SAP tools or using the Dynamic Workload Console.

Jobs in Tivoli Workload Scheduler that correspond to the jobs in SAP
The Tivoli Workload Scheduler job definitions are used in scheduling and defining dependencies, but the SAP jobs are actually run.

You can define SAP job definitions from the Dynamic Workload Console and then have Tivoli Workload Scheduler launch the jobs in SAP R/3 using jobs defined on the following workstations that support the r3batch access method:

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

You can manage your SAP environment from both:

- Distributed ATivoli Workload Scheduler distributed environment
- z/OS A Tivoli Workload Scheduler for z/OS environment.

The SAP job definitions can reference the following types of SAP jobs:

- Standard R/3
- Business Warehouse Process Chains
- Business Warehouse InfoPackages

For information about Business Warehouse Process Chains and Business Warehouse InfoPackages, see "Using Business Information Warehouse" on page 158.

Creating SAP Standard R/3 jobs from the Dynamic Workload Console

How to create and manage an SAP job that is associated to a Tivoli Workload Scheduler job that manages it.

You can easily create and manage Standard R/3 jobs on a remote SAP system entirely from the Dynamic Workload Console, and then continue to manage the remote SAP job from Tivoli Workload Scheduler.

The Tivoli Workload Scheduler job definition, available for both distributed and z/OS environments, maps to the newly created job on the SAP system. The SAP job can run on extended agent workstations, dynamic agent workstations, pools, dynamic pools, and workstations depending on the type of job definition you choose to create.

Note: Using this procedure to create a new Tivoli Workload Scheduler for z/OS Agent SAP Standard R/3 job, you cannot manage variants. To manage variants, use the SAP graphical user interface or use the **List Jobs on SAP** entry from the navigation tree of the Dynamic Workload Console.

To create a new SAP Standard R/3 job on a remote SAP system that maps to a Tivoli Workload Scheduler job definition, you have to associate your SAP Standard R/3 jobs to Tivoli Workload Scheduler jobs and you can do it in either of the following ways:

- Starting from an SAP job: "Create an SAP job and associate it to a Tivoli Workload Scheduler job" on page 107 or
- Starting from a Tivoli Workload Scheduler ("Create a Tivoli Workload Scheduler job and associate it to an SAP job" on page 108)
- Alternatively, you can simply create an SAP job on a remote SAP system, without having it managed by Tivoli Workload Scheduler: "Creating a SAP job from the Dynamic Workload Console" on page 110.

When performing operations that require a connection to a remote SAP system, you must configure the SAP connection data. The connection is made through a Tivoli Workload Scheduler workstation with the r3batch access method installed. Each workstation can have one or more options files that can be used to customize the behavior of the r3batch access method, except for extended agent workstations, where only one options file can be defined and therefore a selection is not required. For information about setting the SAP connection data, see "Setting the SAP data connection" on page 110.

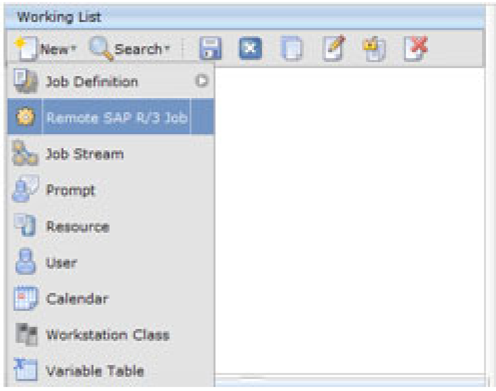
Create an SAP job and associate it to a Tivoli Workload Scheduler job

How to create a new SAP job that is associated to a Tivoli Workload Scheduler job that manages it.

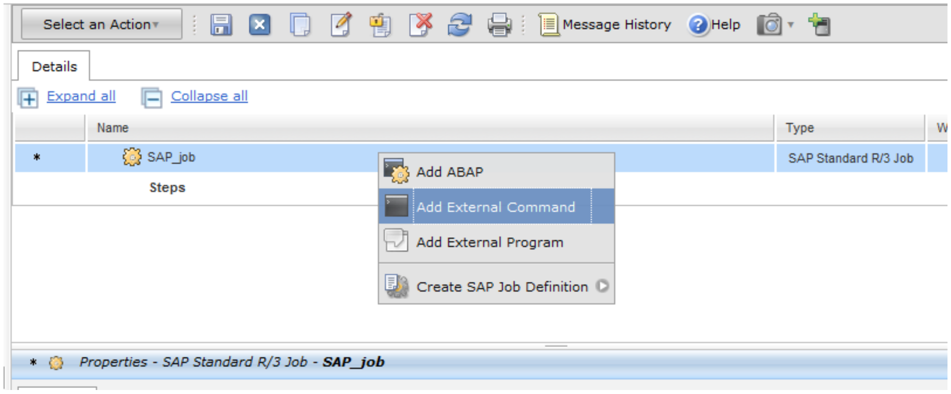
To be able to save your SAP job on a remote SAP system, you must specify the connection details. See "Setting the SAP data connection" on page 110.

To create a new SAP job and then associate it to a new Tivoli Workload Scheduler job, perform the following steps:

1. Click **Workload > Design > Create Workload Definitions**.
2. Select an engine, either distributed or z/OS. The Workload Designer is displayed.
3. From the Working List pane, click **New** > **Remote SAP R/3 Job**:



4. In the Properties pane, specify the properties for the SAP job definition you are creating using the tabs available. The tabs for each type of SAP job definition are similar, but there are some differences depending on the type of engine you selected and the type of workstation on which the job runs. For more detailed information about the UI elements on each tab, see the Dynamic Workload Console online help.
5. Save the job before adding steps to it.
6. Optionally, in the Details view, right-click the new job to add ABAP, External command or External program steps to it:



7. Right-click the SAP job and click **Create SAP Job Definition** to create a new Tivoli Workload Scheduler job associated to the new job on SAP. Select the job definition in accordance with the engine and type of agent on which the job runs.

   SAP For z/OS systems only. This job definition references an existing job on the SAP system and can run on dynamic agent workstations, dynamic pools, and Tivoli Workload Scheduler for z/OS Agent.

   SAP Job on Dynamic Workstations
   For distributed systems only. This job definition can run on dynamic agent workstations, dynamic pools, and Tivoli Workload Scheduler for z/OS Agent workstations.

   SAP Job on XA Workstations
   This job definition can run on extended agent workstations, which are workstations hosted by fault-tolerant agents or master workstations.

8. Right-click the steps to move them and change their sequence.
9. The Tivoli Workload Scheduler job definition opens in the Workload Designer. Some fields in the Properties pane already contain information relating to the associated SAP job. Specify the remaining properties using the tabs available. The tabs for each type of SAP job definition are similar, but there are some differences depending on the type of engine you selected and the type of workstation on which the job runs. For more detailed information about the UI elements on each tab, see the Dynamic Workload Console online help.
10. Click 💾 to save the SAP job definition in the Tivoli Workload Scheduler database.

Create a Tivoli Workload Scheduler job and associate it to an SAP job

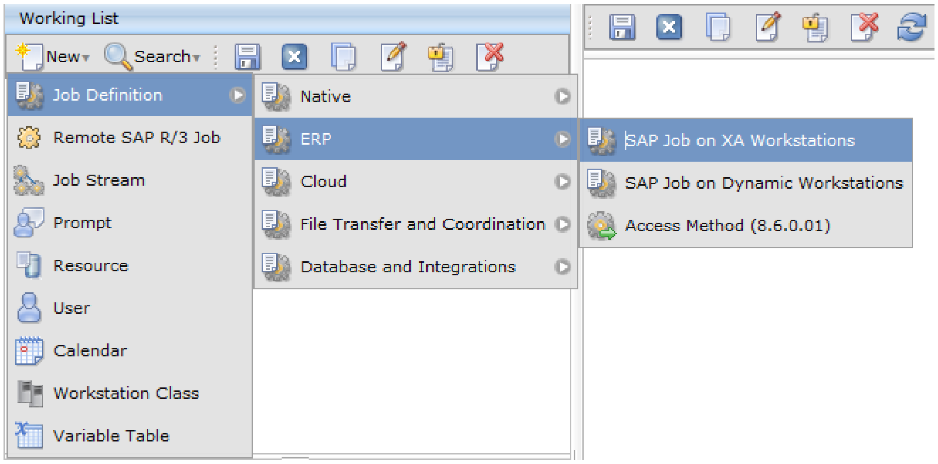
Create a Tivoli Workload Scheduler job definition and map it to a new or existing SAP job to manage it.

To create a new Tivoli Workload Scheduler job and then associate it to a new SAP job, perform the following steps:

1. Click **Workload > Design > Create Workload Definitions**.
2. Select an engine, either distributed or z/OS. The Workload Designer is displayed.
3. From the Working List pane,
   - Distributed Click: **New >Job Definition > ERP > SAP Job on....**, choosing the type of workstation on which it is going to run:

SAP Job on Dynamic Workstations
For distributed systems only. This job definition can run on dynamic agent workstations, dynamic pools, and Tivoli Workload Scheduler for z/OS Agent workstations.

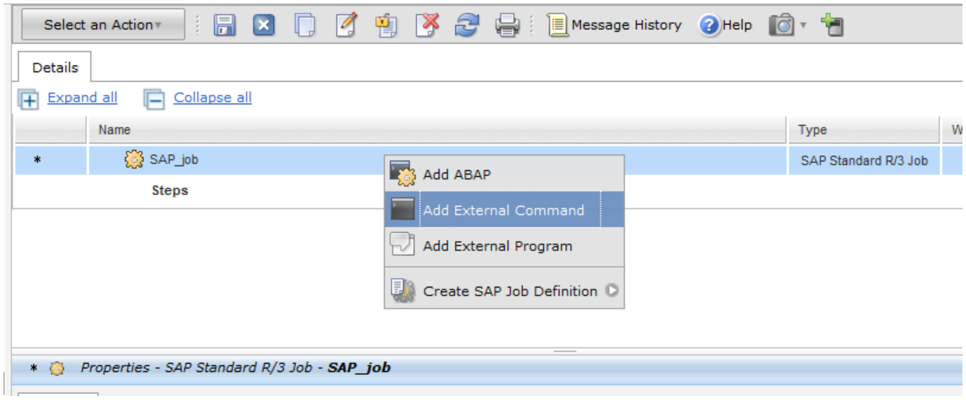
SAP Job on XA Workstations
This job definition can run on extended agent workstations, which are workstations hosted by fault-tolerant agents or master workstations.



- z/OS Click: **New > ERP > SAP**

SAP For z/OS systems only. This job definition references an existing job on the SAP system and can run on dynamic agent workstations, dynamic pools, and Tivoli Workload Scheduler for z/OS Agent.

4. In the Properties pane, specify the properties for the SAP job definition you are creating using the tabs available. The tabs for each type of SAP job definition are similar, but there are some differences depending on the type of engine you selected and the type of workstation on which the job runs. For more detailed information about the UI elements on each tab, see the Dynamic Workload Console online help.
5. In the Task tab, specify the Tivoli Workload Scheduler job that you want to associate to the SAP job. If this job already exists, specify it in the Job name field, otherwise, click **New** to create it from new and specify its properties in the Properties pane.
6. Save the job before adding steps to it.
7. Optionally, in the Details view, right-click the new job to add ABAP, External command or External program steps to it:



8. The Tivoli Workload Scheduler job definition opens in the Workload Designer. Some fields in the Properties pane already contain information relating to the SAP job you have previously created. Specify the remaining properties using the tabs available. The tabs for each type of SAP job definition are similar, but there are some differences depending on the type of engine you selected and the type of workstation on which the job runs. For more detailed information about the UI elements on each tab, see the Dynamic Workload Console online help.
9. Save the job before adding steps to it.

Creating a SAP job from the Dynamic Workload Console

How to create an SAP job definition on a remote SAP system from the Dynamic Workload Console.

You can also create and save SAP Standard R/3 jobs directly on the remote SAP system from Tivoli Workload Scheduler, as you would from the SAP graphical user interface. To create Standard R/3 jobs on the SAP system from the Dynamic Workload Console, perform the following steps:

1. Click **Workload** > **Design** > **List Jobs on SAP**.
2. In the Filter, select **Standard R/3 Job** and specify the workstation name. This parameter is mandatory because it identifies the remote SAP system.
3. Specify the workstation where the SAP job runs. This is the workstation with the r3batch access method that communicates with the remote SAP system.
4. If the workstation is not an extended agent workstation, you must also specify the options file to be used.
5. Click **Display** to view a list of the Standard R/3 jobs for the specified workstation.
6. Click **New** to create a new Standard R/3 job and enter the required information in the **R/3 Job Definition** and **R/3 steps** tabs.
7. Click **OK** to save the job on the SAP system.

After creating the new SAP job on the SAP from the Dynamic Workload Console, you must reference it in a Tivoli Workload Scheduler SAP Standard R/3 job if you want to manage the job from within Tivoli Workload Scheduler as explained in “Create a Tivoli Workload Scheduler job and associate it to an SAP job” on page 108.

Setting the SAP data connection

You can configure a default connection to be used when performing actions that access the remote SAP system.

There are several operations you can perform which require connection details to establish a link to a remote SAP system. The connection is made through a Tivoli Workload Scheduler workstation with the r3batch access method installed used to communicate with the SAP system. Each workstation can have one or more options files that can be used to customize the behavior of the r3batch access method, except for extended agent workstations, where only one options file can be defined and therefore a selection is not required.

For example, you can use the Workload Designer to create Tivoli Workload Scheduler job definitions that reference remote SAP jobs, or you can create a SAP job on a remote SAP system. You can also search for SAP jobs on the remote system from the **Working List** and **Quick Open** panes.

The Workload Designer enables you to configure a default connection that is automatically referenced when performing these types of operations. You can change the default configuration at any time, or you can overwrite the default configuration when performing these operations.

To configure a default SAP data connection to be used when creating objects from the Workload Designer that require a SAP connection, perform the following steps:

1. From the Workload Designer, click from the toolbar of the Details view.
2. In **Workstation**, enter the name of the workstation that communicates with the SAP system or use the pick tool to search for and select one.
3. In **Option file**, enter the options file to be used or use the pick tool to search for options files that reside on the specified workstation and select one.
4. Click **OK**.

A default SAP connection has been configured and will be used each time an object that requires access to a SAP system is defined.

Managing SAP variants using the Dynamic Workload Console

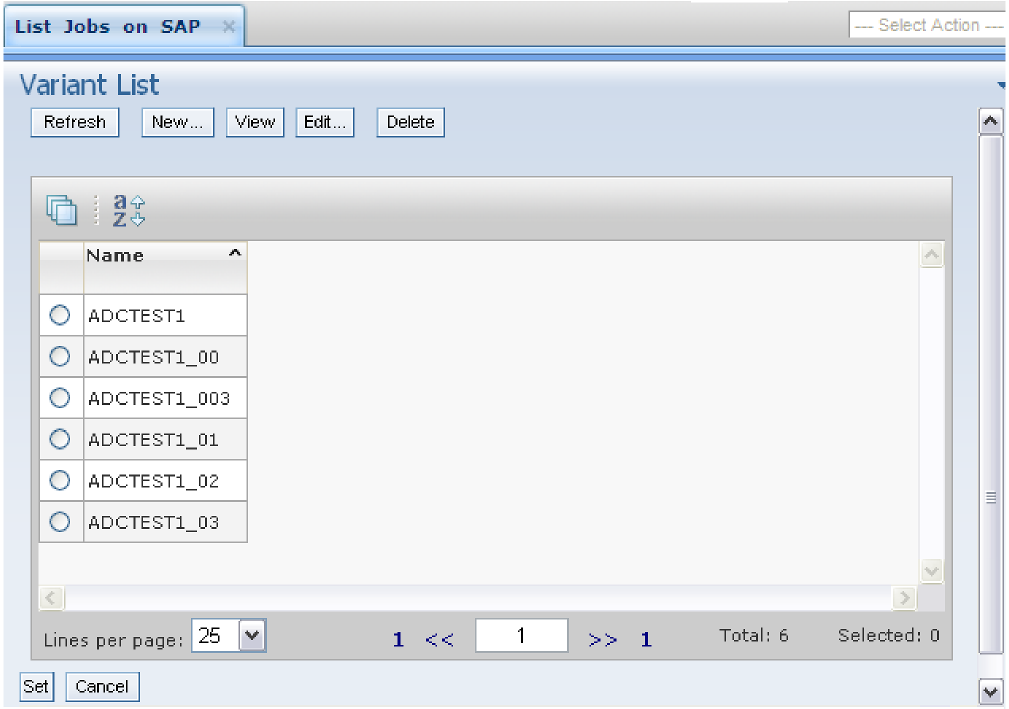
Managing variants using the Dynamic Workload Console.a

This section describes how to manage variants using the Dynamic Workload Console:

1. Click **Workload** > **Design** > **List Jobs on SAP** from the portfolio.
2. Specify an engine connection.
3. In **Workstation name**, type the name of the workstation where the SAP job runs. This is the workstation with the r3batch access method that communicates with the remote SAP system. If you do not know the name of the workstation, click (...) browse to enter your filter criteria and click **Search**. If you enter a string representing part of the workstation name, it must be followed by the asterisk (*) wildcard character. Both the question mark (?) and asterisk (*) are supported as wildcards. You can also simply use the asterisk wildcard character (*) to display all workstations. Optionally, specify any of the other search criteria available and click **Search**. From the results displayed, select the workstation and click **OK**.
4. In **Option file**, specify an options file that resides on the specified workstation. Each workstation can have one or more options files that can be used to customize the behavior of the r3batch access method, except for extended agent workstations, where only one options file can exist and therefore does not need to be specified. For the workstation specified, enter

the file name of the options file or click the browse (...) button to search for options files that reside on the specified workstation and select one.

5. Click **Display**. The list of available jobs on the remote SAP system for the specified engine is displayed.
6. A list of SAP jobs on the remote SAP system are displayed.
7. Select a SAP job from the list and click **Edit**.
8. On the **R/3 Steps** page, select a program of type ABAP from the list and click **Edit**. The properties for the ABAP program are displayed.
9. In the **Variant** field, click the ellipsis (**...**) icon to display the Variant List panel. This panel lists all the variants associated with the ABAP specified in the **Name** field.



	Name
○	ADCTEST1
○	ADCTEST1_00
○	ADCTEST1_003
○	ADCTEST1_01
○	ADCTEST1_02
○	ADCTEST1_03

Lines per page: 25 1 << 1 >> 1 Total: 6 Selected: 0

Figure 5. The Variant List panel

10. From this panel, you can perform the following actions:

Refresh
To refresh the content of the variant list with the information contained in the SAP database.

New To create a new variant as described in “Creating or editing a variant” on page 113.

View To display information on an existing variant.

Edit To modify information on an existing variant as described in “Creating or editing a variant” on page 113.

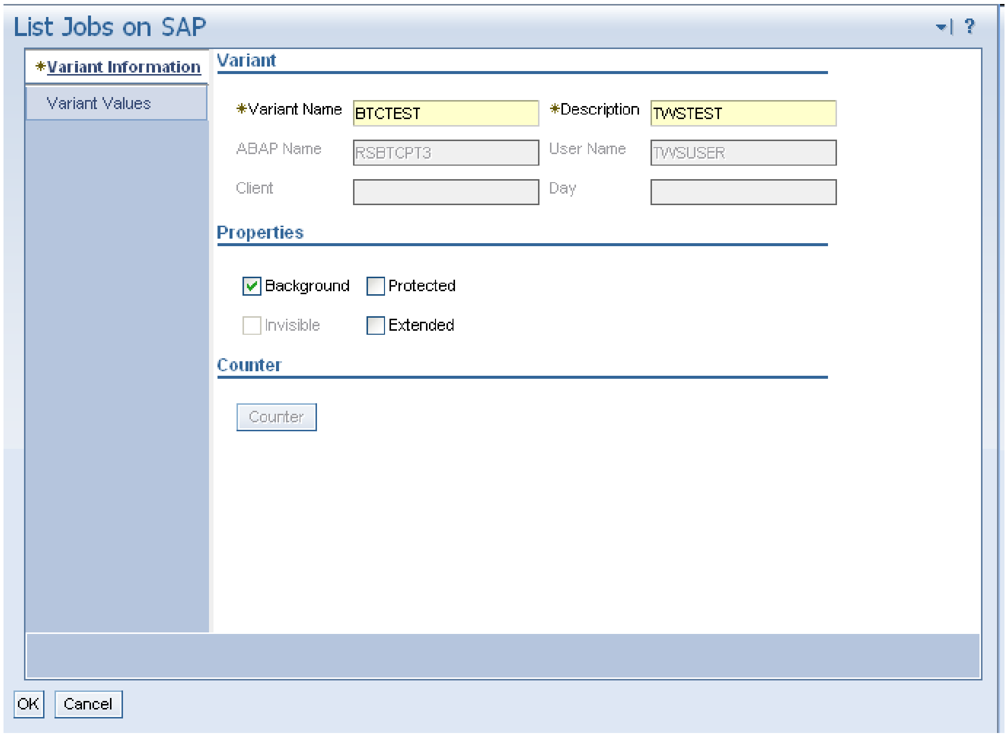
Delete To delete a variant.

Set To associate the value chosen from the list to the ABAP.

Creating or editing a variant

You can create or edit a variant from the **Variant List** panel. To display the **Variant List** panel, see "Managing SAP variants using the Dynamic Workload Console" on page 111.

1. In the Variant List panel, click **New** or **Edit**. The Variant Information page is displayed by default. If you are editing an existing variant, the fields and selections are not empty.



Variant

*Variant Name: BTCTEST
*Description: TWSTEST
ABAP Name: RSBTCPT3
User Name: TWSUSER
Client:
Day:

Properties

☑ Background ☐ Protected
☐ Invisible ☐ Extended

Counter

Figure 6. The Variant Information page of the Variant List panel

2. The panel consists of the following pages:
   - Variant Information:
     a. Enter or modify the variant name and description.
     b. Optionally, check a **Properties** box:

        Background
        The variant can only be used in background processing.

        Protected
        The variant is protected against being changed by other users.

        Invisible
        The variant will not be displayed in the F4 value list on the SAP GUI. Not available for the BC-XBP 3.0 interface.

        Extended
        Allows for the use of placeholders and counters as variant values. If you check this box, **Counter** becomes available.

        For extended variants, you can use placeholders and counters that eliminate the error-prone task of adjusting values and therefore minimize the effort for variant maintenance. Placeholders and counters are preprocessed by Tivoli Workload Scheduler and the values are automatically adjusted when the job is launched. Supported placeholders and counters are:

Table 36. Placeholders and counters for extended variants

Symbol	Meaning	Syntax
$S	Timestamp	YYYYMMDDHHMM
$D	Day of the month	DD
$_D	Date	YYYYMMDD
$M	Month	MM
$Y	Year	YY
$_Y	Year	YYYY
$H	Hour	HH
$T	Minute	MM
$_T	Time	HHMMSS
$N*x*	Counters	10 counters: $N0 - $N9 ($N = $N0)
$(*date expression*)	Date expression	Like the `datecalc` command. Enclosed within $(and).
$[*arithmetic expression*]	Arithmetic expression	Arithmetic expressions allowing for +, -, *, and / operations between integers and counters.

- Variant Values:

  In the Variant Values page, the fields and values are dynamically built through `r3batch` depending on the characteristics of the variant or step and are identical to the ones in the equivalent SAP panel.

Editing a standard SAP job

There are two ways in which you can edit SAP Standard R/3 jobs from Tivoli Workload Scheduler.

- The Dynamic Workload Console contains the **List Jobs on SAP** entry in the portfolio for creating and editing SAP Standard R/3 jobs on remote SAP systems.
- The Workload Editor from the Dynamic Workload Console enables you to create and edit remote SAP jobs. See “Creating SAP Standard R/3 jobs from the Dynamic Workload Console” on page 106.

To edit a SAP standard R/3 job, perform the following steps:

1. Open the **Workload Designer**, from the portfolio, click **Tivoli Workload Scheduler** > **Workload** > **Design** > **List Jobs on SAP**.
2. In **Engine name**, select the name of the Tivoli Workload Scheduler engine connection from which you want to work with SAP jobs.
3. Leave the default setting in the **SAP Job Type** section to **Standard R/3 Job**.
4. In **Workstation name**, type the name of the workstation where the SAP job runs. This is the workstation with the r3batch access method that communicates with the remote SAP system. If you do not know the name of the workstation, click (...) browse to enter your filter criteria and click **Search**. If you enter a string representing part of the workstation name, it must be followed by the asterisk (*) wildcard character. Both the question mark (?) and

asterisk (*) are supported as wildcards. You can also simply use the asterisk wildcard character (*) to display all workstations. Optionally, specify any of the other search criteria available and click **Search**. From the results displayed, select the workstation and click **OK**.

5. In **Option file**, specify an options file that resides on the specified workstation. Each workstation can have one or more options files that can be used to customize the behavior of the r3batch access method, except for extended agent workstations, where only one options file can exist and therefore does not need to be specified. For the workstation specified, enter the file name of the options file or click the browse (...) button to search for options files that reside on the specified workstation and select one.
6. Click **Display**. The list of available jobs on the remote SAP system for the specified engine is displayed.
7. Select the job you want to modify in the list and click **Edit**. The List Jobs on SAP panel is displayed.
8. Edit the properties on the **R/3 Job Definition** and **R/3 Steps** pages as appropriate. Refer to the contextual online help available for more detailed information about the UI elements available on each page.

   Note:

   - On the **R/3 Job Definition** page, when you modify the **Job Class**, **Target Host**, or **Server Group** and click **OK**, the **Job ID** is maintained and remains synchronized with the one associated to the current job. Instead, when you modify the **Job Name** and click **OK**, the **Job ID** is automatically replaced with the one associated to the new job name.
   - On the **R/3 Steps** page, for each step you modify, the new step information is saved in the SAP database. For each step you add or delete, the **Job ID** is maintained and remains synchronized with the one associated to the modified step.
9. Click **OK** to save your changes.

Task string to define SAP jobs

This section describes the task string parameters that define and control the running of SAP jobs. You can specify them in the following places when you define their associated Tivoli Workload Scheduler jobs:

- In the **R/3 Command Line** section of the Task page of the Submit Ad Hoc Jobs action from the Dynamic Workload Console.
- In the **R3 Command Line** field of the More Options page of the SAP job definition, if you use the Dynamic Workload Console and selected a **SAP** job definition.
- As arguments of the `scriptname` keyword in the job definition statement, if you use the Tivoli Workload Scheduler command line.
- As arguments of the `JOBCMD` keyword in the `JOBREC` statement in the `SCRIPTLIB` of Tivoli Workload Scheduler for z/OS, if you are scheduling in an end-to-end environment. The following is an example of a `JOBREC` statement:

  ```
  JOBREC
    JOBCMD('/-job job_name -user user_name -i job_ID -c class_value')
    JOBUSR(TWS_user_name)
  ```

  where:

class_value
: The priority with which the job runs in the SAP system. For details, see Table 37 on page 118.

job_ID The unique SAP job ID. For details, see Table 37 on page 118.

job_name
: The name of the SAP job to run. For details, see Table 37 on page 118.

user_name
: The SAP user who owns the target job. For details, see Table 37 on page 118.

TWS_user_name
: The Tivoli Workload Scheduler for z/OS user who runs the `r3batch` access method from the end-to-end scheduling environment.

The string syntax is the following:

Job definition syntax

```
►►── -job──job_name──┬────────────────────┬──┬───────────────────┬───────────────►
                     └─┬─ -i──┬──job_ID─┘  └─ -user──user_name─┘
                       └─ -id─┘

►──┬───────────────────────┬──┬───────────────────────┬─────────────────────────►
   └─┬─ -host─┬──host_name─┘  └─ -sg──server_group─┘
     └─ -ts───┘

►──┬──────────────────────────┬──┬──────────────────────────────┬───────────────►
   └─ -client──source_client─┘  └─ -exec_client──target_client─┘

►──┬─────────────────────────────────┬──┬──────────────────┬────────────────────►
   └─ -rfc_client──rfc_logon_client─┘  └─ -c──class_value─┘

►──┬──────────────────────────────────────────────┬──┬────────────────┬─────────►
   └─ -bdc_job_status_failed──bdc_processing─┘  └─┬─ -nobdc─────┤
                                                     └─ -nobdcwait─┘

►──┬─────────────────────────────────────┬──┬───────────────────────────┬───────►
   │                      ┌─high───┐     │  └─ -s──starting_step_number─┘
   └─ -bapi_sync_level───┼─medium─┼────┘
                          └─low────┘
   ┌───────────────────────────────────────────────────────┐
   ▼                                                       │
►──┬───────────────────────────────────────────────────────┬──────────────►
   └─ -sStep_number─attribute_name──┬────────────────────┬─┘
                                    └─=attribute_value───┘

►──┬──────────────────────────────┬──┬──────────────────────────────────────────┬──►
   └─ -vStep_number─variant_name──┘  └─ -vtxtStep_number─variant_description────┘

►──┬──────────────────────────────────────────────┬───────────────────────────────►
   └─ -vparStep_number─name=variant_value─────────┘

►──┬───────────────────────────────────────────────────────────────────────┬──────►
   └─ -vselStep_number─name=─┬─i─┬─#operation#lowest─┬─────────────────┬───┘
                             └─e─┘                   └─#highest────────┘

►──┬─────────────────────┬──┬──────────────────────────────┬──────────────────────►
   └─ -vtempStep_number──┘  └─ -recipient──R/3_login_name──┘

►──┬──────────────────────────────┬──┬──────────────────────┬──┬──────────────────┬──►
   └─ -rectype──recipient_type────┘  └─ -flag──┬─reccp─┬────┘  └─ -flag recex─────┘
                                               └─recbl─┘

►──┬────────────────┬──┬───────────────────────┬──┬───────────────────────────────────┬──►
   └─ -flag recnf───┘  └─ -flag──┬─im─────┬────┘  └─ -flag──┬─enable_applinfo───┬────┘
                                 └─immed──┘                 └─disable_applinfo──┘

►──┬─────────────────────────────────────┬──┬─────────────────────────────────┬──►
   └─ -flag──┬─enable_appl_rc───┬────────┘  └─ -flag──┬─enable_joblog───┬─────┘
             └─disable_appl_rc──┘                     └─disable_joblog──┘

►──┬─────────────────────────────────────────────────┬──┬───────────────────────────────────┬──►
   └─ -flag──┬─enable_job_interceptable───┬──────────┘  └─ -flag──┬─enable_spoollist───┬────┘
             └─disable_job_interceptable──┘                       └─disable_spoollist──┘

►──┬─────────────────────┬──┬──────────┬──┬─────────────────────┬──┬─────────────┬──►◄
   └─ -flag──pc_launch───┘  └─ -debug──┘  └─ -tracelvl──┬─1─┬───┘  └─ -rfctrace──┘
                                                        ├─2─┤
                                                        └─3─┘
```

Table 37 on page 118 describes the parameters for the task string to define SAP jobs.

Note:

1. You can specify both `-i` or `-id` and `-user` in the same job definition, but the user name is ignored.
2. When you specify the job ID, both `-client` and `-exec_client` are ignored because the ID is unique for the entire SAP system.
3. Typically, the `-debug` and `-trace` options are for debugging the extended agent and should not be used in standard production.

Table 37. Task string parameters for SAP jobs

Section	Parameters	Description	GUI Support
JOB	-job *job_name*	The name of the job to run. This parameter is mandatory.	✓
	-i *job_ID* -id *job_ID*	The unique SAP job ID. Specify this parameter if you are submitting a job that refers to a predefined job template stored in the SAP database for which you want to change a parameter.	✓
	-user *user_name*	The SAP user who owns the target job. Use this parameter when the target SAP system has only one job with the specified name for the specified user. This parameter has no effect if a job ID is specified in the job definition.	✓
	-host *host_name* -ts *host_name*	The name of the SAP workstation where the job is to be run. *host_name* has the format *hostname_SAPsystemname_SAPsystemnumber*. For example, the name of a host might be amss80a0_gs7_90 These parameters are mutually exclusive with -sg.	✓
	-sg *server_group*	The name of the SAP server group where the job is to be run. Use this parameter to run the job on an application server that belongs to the group. The server group must exist on the SAP system, otherwise an error code is returned and the job is not launched. This parameter is case-sensitive and can be up to 20 characters. It is mutually exclusive with -host and -ts.	✓

Table 37. Task string parameters for SAP jobs (continued)

Section	Parameters	Description	GUI Support
JOB	-client *source_client*	The number that identifies the SAP client where the job definition is to be found, regardless of the client number defined by the `r3client` keyword in the options file. This parameter has no effect if a job ID is specified in the job definition.	
	-exec_client *target_client*	The number that identifies the SAP client where the job is to be run, regardless of the client number defined by the r3client keyword in the options file. This parameter has no effect if a job ID is specified in the job definition.	
	-rfc_client *rfc_logon_client*	The number that identifies the SAP client to be used for RFC logon. This value overwrites the value specified by the r3client keyword in the corresponding r3batch options file.	
	-c *class_value*	The priority with which the job runs in the SAP system. Possible values are: **A** High priority **B** Medium priority **C** Low priority. This is the default value.	✓
	-bdc_job_status_failed *bdc_processing*	How Tivoli Workload Scheduler sets the completion status of a job running BDC sessions, according to a possible BDC processing failure. The allowed values are: *n* If at least *n* BDC sessions failed (where *n* is an integer greater than 0), Tivoli Workload Scheduler sets the job completion status as failed. **all** If all the BDC sessions failed, Tivoli Workload Scheduler sets the job completion status as failed. **ignore** When all the BDC sessions complete, regardless of their status, Tivoli Workload Scheduler sets the job completion status as successful. This is the default. If **-nobdc** or **-nobdcwait** is set, this option is ignored.	✓
	-nobdc -nobdcwait	Disables the BDC Wait option (enabled by default) to have the job considered as completed even if not all its BDC sessions have ended.	✓
	-bapi_sync_level	Specifies the synchronization level between the SAP function modules BAPI_XBP_JOB_COPY and BAPI_XBP_JOB_START_ASAP. Allowed values are: **high** All RFC calls between BAPI_XBP_JOB_START_ASAP and BAPI_XBP_JOB_COPY are synchronized. This is the default. **medium** The RFC calls to BAPI_XBP_JOB_START_ASAP are synchronized. **low** The RFC calls are not synchronized.	

Table 37. Task string parameters for SAP jobs (continued)

Section	Parameters	Description	GUI Support
STEP	-s *starting_step_number*	The number of the starting step.	✓

	-s*Step_number attribute_name=attribute_value*	The step number and its attributes, where:
		step_number
		The number of the step being defined. Each step is identified by a sequential number (1, 2, 3, ...*n*) using the *step number*
		attribute_name
		The name of the attribute.
		attribute_value
		The value of the attribute. It is optional for some attributes.
		Attributes can be defined in any order, but cannot be repeated for the same step. Attribute validation is performed before the job is created in the SAP system. If the validation fails, the Tivoli Workload Scheduler job goes into the ABEND state. For a detailed description of each attribute and its value, see "Defining attributes for ABAP steps" on page 140 and "Defining attributes for external programs and external commands steps" on page 142.
		For example, the following step (step 8) is an ABAP module running the report MYPGM and has two attributes, only one of which has a value.
		`-s8 type=A -s8 program=MYPGM`
		`-s8 pr_cover="My title" -s8 pr_immed`
			✓

Table 37. Task string parameters for SAP jobs (continued)

Section	Parameters	Description	GUI Support
VARIANT	-v*step_number* name	The variant name for the specified step number.	✓
	-vtxt*step_number variant_description*	The textual description of the variant, in the Tivoli Workload Scheduler logon language (customizable with the `TWSXA_LANG` option of `r3batch`). The maximum length is 30 characters.	✓
	-vpar*step_number* name=*value*	For ABAP modules only. The value for a variant parameter for the specified step number. This parameter is mandatory when creating a new variant. See "Defining attributes for ABAP steps" on page 140 for a complete list of the supported attributes for ABAP steps.	✓
	-vsel*step_number* name=*sign*#*operation*#*lowest*[#*highest*]	For ABAP modules only. The value for a variant selection option for the specified step number. **sign** Sign of the operation. Possible values are: **I** Include **E** Exclude **operation** Possible values are: **EQ** Equals **NE** Not equal to **BT** Between **NB** Not between **LT** Less than **LE** Less than or equal to **GT** Greater than **GE** Greater than or equal to **CP** Contains pattern **NP** Does not contain pattern **lowest** Low value of the selection. You can use up to 45 characters. **highest** High value of the selection. You can use up to 45 characters. This attribute is optional. For a complete list of the supported attributes for ABAP steps, see"Defining attributes for ABAP steps" on page 140.	✓
	-vtemp*step_number*	For ABAP modules only. Specifies to assign a temporary variant to the specified step number. Temporary variants are created ad-hoc by the SAP system and assigned to the job instance when it is run. The lifecycle of the temporary variant is determined by the SAP system. If the job is deleted by SAP, then the temporary variant is deleted. See "Examples: Dynamically defining and updating SAP jobs" on page 144 to refer to examples that demonstrate the behavior of temporary variants.	

Table 37. Task string parameters for SAP jobs (continued)

Section	Parameters	Description	GUI Support
SPOOL	-recipient *name*	The login name of an SAP user.	
	-flag {reccp | recbl}	Specifies how the spool list is sent to the recipient. Possible values are: **reccp** The spool list is sent as a copy. **recbl** The spool list is sent as a blind copy.	
	-flag recex	Specifies that the spool list is sent as an express message to the recipient.	
	-flag recnf	Specifies that the recipient is not allowed to forward the spool list.	
	-rectype type	Specifies the recipient type. Possible values are: **' '** SAP user (default value) **'B'** SAP user **'C'** Shared distribution list **'D'** X.500 address **'G'** Organization object/ID **'H'** Organization unit **'I'** SAP object **'L'** Telex number **'O'** SAPoffice user **'P'** Private distribution list **'R'** SAP user in another SAP system **'U'** Internet address **'1'** Other recipient type	
FLAGS	-flag im -flag immed	Specifies to launch job immediately, meaning that if there are no spare work processes, the job fails.	✓
	-flag enable_applinfo -flag disable_applinfo	Enables or disables the retrieval and appending of the SAP application log to the stdlist of Tivoli Workload Scheduler.	✓
	-flag enable_appl_rc -flag disable_appl_rc	Enables or disables the mapping of the SAP application return code to the Tivoli Workload Scheduler return code.	
	-flag enable_joblog -flag disable_joblog	Enables or disables retrieval of the joblog.	✓
	-flag enable_job_interceptable -flag disable_job_interceptable	Enables or disables the job launched by r3batch to be intercepted by SAP. If enabled, when r3batch launches a job and the SAP job interception feature is enabled, the job can be intercepted if it matches previously defined criteria. If disabled, the job launched by r3batch cannot be intercepted by SAP. This setting overwrites the setting in the common options file.	✓
	-flag enable_spoollist -flag disable_spoollist	Enables or disables retrieval of the spool lists of the job.	✓

Table 37. Task string parameters for SAP jobs (continued)

Section	Parameters	Description	GUI Support
-flag pc_launch	Specifies to launch child jobs that are in scheduled state. **ON** The product launches child jobs that are in scheduled state. **OFF** The product does not launch child jobs that are in scheduled state. This is the default value. **Note:** You can use this option only if you activated the parent-child feature on the SAP system. On the XBP 2.0 (or later)SAP system you can activate this feature using the INITXBP2 ABAP report		
TRACING	-debug	Enables maximum trace level.	✓
	-tracelvl 1|2|3	Specifies the trace setting for the job. Possible values are: **1** Only error messages are written in the trace file. This is the default. **2** Informational messages and warnings are also written in the trace file. **3** A most verbose debug output is written in the trace file. For detailed information, refer to “Configuring the tracing utility” on page 281.	✓
	-rfctrace -trace	Enables RFC trace.	

The following is an example for an SAP job named `BVTTEST` with ID `03102401` and user `myuser`:

```
-job BVTTEST –i 03102401 -user myuser -debug
```

Managing SAP jobs

This chapter describes how to manage SAP jobs.

Displaying details about a standard SAP job

Perform the following steps to display details for standard jobs on specific workstations.

For information about how to display details about a job that submits an SAP process chain, refer to “Displaying details about a process chain job” on page 164.

1. Open the **Workload Designer**, from the portfolio, click **Tivoli Workload Scheduler** > **Workload** > **Design** > **List Jobs on SAP**.
2. In **Engine name**, select the name of the Tivoli Workload Scheduler engine connection from which you want to view SAP job details.
3. In **Workstation name**, type the name of the workstation where the SAP job runs. This is the workstation with the r3batch access method that communicates with the remote SAP system. If you do not know the name of

the workstation, click (...) browse to enter your filter criteria and click **Search**. If you enter a string representing part of the workstation name, it must be followed by the asterisk (*) wildcard character. Both the question mark (?) and asterisk (*) are supported as wildcards. You can also simply use the asterisk wildcard character (*) to display all workstations. Optionally, specify any of the other search criteria available and click **Search**. From the results displayed, select the workstation and click **OK**.

4. In **Option file**, specify an options file that resides on the specified workstation. Each workstation can have one or more options files that can be used to customize the behavior of the r3batch access method, except for extended agent workstations, where only one options file can exist and therefore does not need to be specified. For the workstation specified, enter the file name of the options file or click the browse (...) button to search for options files that reside on the specified workstation and select one.
5. Click **Display**. The list of available jobs for the specified engine is displayed.
6. Select the job for which you want to display the details and click **Details**. The List Jobs on SAP panel is displayed containing job and time information.
7. When you have finished viewing the details for the job, click **OK** to return to the list of SAP jobs on the workstation specified.

Verifying the status of a standard SAP job

To verify the status of a standard SAP job, perform the following steps:

1. Open the **Workload Designer**, from the portfolio, click **Tivoli Workload Scheduler** > **Workload** > **Design** > **List Jobs on SAP**.
2. In **Engine name**, select the name of the Tivoli Workload Scheduler engine connection from which you want to verify the status of an SAP job.
3. In **Workstation name**, type the name of the workstation where the SAP job runs. This is the workstation with the r3batch access method that communicates with the remote SAP system. If you do not know the name of the workstation, click (...) browse to enter your filter criteria and click **Search**. If you enter a string representing part of the workstation name, it must be followed by the asterisk (*) wildcard character. Both the question mark (?) and asterisk (*) are supported as wildcards. You can also simply use the asterisk wildcard character (*) to display all workstations. Optionally, specify any of the other search criteria available and click **Search**. From the results displayed, select the workstation and click **OK**.
4. In **Option file**, specify an options file that resides on the specified workstation. Each workstation can have one or more options files that can be used to customize the behavior of the r3batch access method, except for extended agent workstations, where only one options file can exist and therefore does not need to be specified. For the workstation specified, enter the file name of the options file or click the browse (...) button to search for options files that reside on the specified workstation and select one.
5. Click **Display**. The list of available jobs for the specified engine is displayed.
6. Select the job for which you want to verify the status and click **Status**. The current status for the SAP job is displayed, as well as the database name where the job is installed.
7. When you have finished verifying the status for the job, click **OK** to return to the list of SAP jobs on the workstation specified.

Deleting a standard SAP job from the SAP database

To delete a standard SAP job from the SAP database, perform the following steps:

1. Open the **Workload Designer**, from the portfolio, click **Tivoli Workload Scheduler** > **Workload** > **Design** > **List Jobs on SAP**.
2. In **Engine name**, select the name of the Tivoli Workload Scheduler engine connection from which you want to delete the SAP job.
3. In **Workstation name**, type the name of the workstation where the SAP job runs. This is the workstation with the r3batch access method that communicates with the remote SAP system. If you do not know the name of the workstation, click (...) browse to enter your filter criteria and click **Search**. If you enter a string representing part of the workstation name, it must be followed by the asterisk (*) wildcard character. Both the question mark (?) and asterisk (*) are supported as wildcards. You can also simply use the asterisk wildcard character (*) to display all workstations. Optionally, specify any of the other search criteria available and click **Search**. From the results displayed, select the workstation and click **OK**.
4. In **Option file**, specify an options file that resides on the specified workstation. Each workstation can have one or more options files that can be used to customize the behavior of the r3batch access method, except for extended agent workstations, where only one options file can exist and therefore does not need to be specified. For the workstation specified, enter the file name of the options file or click the browse (...) button to search for options files that reside on the specified workstation and select one.
5. Click **Display**. The list of available jobs for the specified engine is displayed.
6. Select the job or jobs you want to delete and click **Delete**. A confirmation message prompts you to confirm the delete action.
7. When the delete action is complete, click **OK** to return to the list of SAP jobs on the workstation specified.

Balancing SAP workload using server groups

SAP jobs run on application servers that host work processes of type *batch*. Critical batch jobs are run in specific time frames, on specific application servers. With SAP Basis version 6.10 and later, application servers can be assigned to server groups. With Tivoli Workload Scheduler for Applications, you can assign a server group to a job. In this way, when a job is launched, the SAP system runs it on an application server that belongs to that group, balancing the workload among the various application servers in the group.

If the application servers defined in a group are modified in the SAP system, the job defined as belonging to that server group is not affected and does not need to be modified. The batch execution targets are reorganized in the SAP system without having to change job definitions in Tivoli Workload Scheduler for Applications.

This function is supported with the following versions of SAP:
- SAP Basis 6.10, with Service Pack 40
- SAP Basis 6.20, with Service Pack 41
- SAP Basis 6.40, with Service Pack 04
- SAP Basis 7.00, and later

Mapping between Tivoli Workload Scheduler and SAP job states

When an SAP job is launched by Tivoli Workload Scheduler, you can monitor its progress. The status transitions in Tivoli Workload Scheduler (internal status) and the corresponding SAP status are listed in Table 38.

Table 38. Status transitions in Tivoli Workload Scheduler (internal status) and the corresponding SAP R/3 status

Tivoli Workload Scheduler Job State	SAP Job State
INTRO	Not Available
WAIT	Ready, Release
EXEC	Active
SUCC	Finished
ABEND	Canceled

The INTRO state indicates that Tivoli Workload Scheduler is in the process of introducing the job, but in SAP, the job has not yet entered the ready state. Because it takes some time to get a job queued and into the ready column, the INTRO state might last a few minutes if the SAP system is particularly busy.

Even if a job is finished in SAP, Tivoli Workload Scheduler keeps it in the EXEC state if its BDC sessions are not complete and you have not selected the **Disable BDC Wait** option. For details about this option, see "Using the BDC Wait option" on page 149.

Managing spools

Browse spool lists on request without having to download the entire spool which can occupy significant space on the file system.

Spool lists can be very large so rather than download them as part of a job run, you can request to browse the spool list, chunks at a time, even if you have disabled the option, `retrieve_spoollist`, to append the spool list to the Tivoli Workload Scheduler joblog.

From the Dynamic Workload Console, you can list the spool data available for SAP Standard R/3 jobs that have run. Each spool is identified by the following information:

- The spool number.
- The related step number.
- The name of the spool request.
- The title of the spool request.
- The total number of pages for the spool information.
- The user who executed the SAP job related to the spool.
- The date the spool was created based on the Coordinated Universal Time (UTC) time standard.
- The client for which the spool was created.

Browsing spool data

You can list the spool data available for SAP Standard R/3 jobs that have run and browse the contents of the spool.

To browse spool data for a specific job that has run:

1. From the portfolio click, **Tivoli Workload Scheduler** > **Workload** > **Monitor** > **Monitor Jobs**
2. Select a task name in the **Task Name** column.
3. Select an engine on which the monitor task must run.
4. Click **OK**.
5. Enter the User ID and Password for the specified engine if required.
6. Select a SAP Standard R/3 job from the list and click **More Actions** > **Show Spool List**. The list of spool data available for the selected job is displayed.
7. Select a spool and click **Spool**.

By default, the first ten pages of the spool are made available. You can change this default by editing the number of pages specified in **Pages for screen**. Use the page functions to jump to a specific page number, jump to the last page of the spool, jump to the first page of the spool, or move forward or back through the number of pages indicated by **Pages for screen**.

Killing an SAP job instance

This section describes how to kill a Tivoli Workload Scheduler job that submits either a standard SAP job or an SAP process chain.

Note: To perform this function on a Windows operating system, you must install IBM Tivoli Workload Scheduler version 8.2.1 Fix Pack 1, or later.

To kill an SAP job instance, perform the following steps:

The Tivoli Workload Scheduler job status is set to ABEND. The SAP job or process chain is set to canceled in the SAP system.

Note: If you kill a process chain job, the SAP system stops as soon as the process that is currently running completes.

1. Display a list of defined job instances containing the job you want to kill. From the Dynamic Workload Console portfolio click **Workload** > **Monitor** > **Monitor Jobs**. If you have a specific montoring task defined, run the task. Refer to the *Dynamic Workload Console User's Guide* for information about creating a task to monitor jobs.
2. The **Monitor Jobs** panel is displayed. Click the task in the **Task Name** column for the plan containing the job instance you want to kill. The task lists all the job instances in the plan.
3. Select the engine associated to the job instance you want to list and click **OK**. The list of job instances associated with the specified engine is displayed.
4. Select the job instance in the list and click **More Actions** > **Kill**.

Raising an SAP event

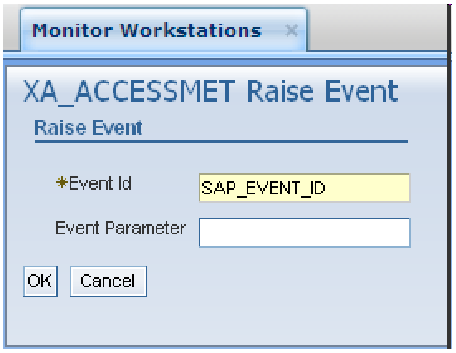
You can raise events on XBP 2.0 (or later) SAP jobs in the Tivoli Workload Scheduler database in one of the following ways:

Using the Monitor Workstations entry from the Dynamic Workload Console

Perform the following steps:

1. On the SAP system, create a job that has as start condition a SAP event. When you create this job, its status is released.
2. Check that this job was not intercepted by the interception function.

3. From the Dynamic Workload Console portfolio, expand **Scheduling Environment** > **Monitor Workstations**.
4. Select the monitoring task to display the list of workstations you want to monitor.
5. In the **Engine name** field, specify the engine connection to be used for this monitoring task and click **OK**. A list of workstations is displayed.
6. Select a workstation that has been defined to connect to a remote SAP system.
7. From the toolbar, select **More Actions** > **Raise Event**. The Raise Event panel opens.



*Event Id: SAP_EVENT_ID

Figure 7. The Raise Event panel

8. The panel consists of the following:

   Event ID
   The identifier of the event that is to be raised.

   SAP Event Parameter
   The parameter of the event that is to be raised.
9. Click **OK**. The event is raised.

Creating a job that launches a Windows or UNIX command that performs a raise event

Perform the following steps:

1. From the Dynamic Workload Console portfolio, expand **Workload** > **Design** > **Create Workload Definitions**.
2. Specify an engine name, either distributed or z/OS. The Workload Designer opens. Job types and characteristics vary depending on whether you select a distributed or a z/OS engine.
3. In the Working List pane, select **New** > **Job Definition** .
4. Select the **Native** category and then either **Windows** or **UNIX**.
5. Use the **General** page to provide general information about the new job definition.
6. Use the **Task** page to provide task information for the job.
7. In the **Task** page, select **Command** and type the following command that performs the raise event in the command string:

```
TWS_home/methods/r3event -c workstation_name -u user_name
-e SAP_event_ID -p parameter
```

   where:

workstation_name
The name of the workstation where the SAP R/3 job is defined.

user_name
The name of the SAP user with which the access method connects to the SAP system. This is the name specified in the r3user option.

SAP_event_ID
The identifier of the event.

parameter
The parameter defined for the event.

8. Save the job definition.

See “Defining conditions and criteria” on page 145 for information about how to define criteria that manages which raised events to log.

Rerunning a standard SAP job

You can rerun a standard SAP job from the start, or from a specific numeric step of the SAP instruction.

To rerun a standard SAP job, you can use one of the following user interfaces:

conman
For details, refer to the *Tivoli Workload Scheduler: User's Guide and Reference*.

Dynamic Workload Console
Dynamic Workload Console

For details about how to rerun a job that submits an SAP process chain, refer to “Rerunning a process chain job” on page 167.

For an SAP extended agent, a `step` is the numeric step of the SAP instruction from which a job can be restarted. Before you rerun an SAP job with Tivoli Workload Scheduler, you have the option of providing a step name for the job. This affects `r3batch` in the following ways:

- If you use a step name that is up to 9 digits (or 8 digits preceded by a character) in length, this name is used as the starting step number for the rerunning job.
- If you use any different format, the name is ignored and the job is rerun starting from the first step.

For example, to rerun a job from the third step, you can use: `A03`, `3`, `00003`, or `H3`.

z/OS In z/OS environments, you need to set the status of the job to **Ready** before you can rerun the job.

Note: By default, if you specify a job step to rerun, the new job is assigned the name of the step you indicated. To keep the original job name, set the Tivoli Workload Scheduler global option **enRetainNameOnRerunFrom** to `yes`. For details about this option, see *Tivoli Workload Scheduler: Planning and Installation*.

When `r3batch` reruns a job from its first step, either because you specified it as the starting step or because no starting step was specified, it uses the `new copy` feature, if applicable. If the starting step is greater than one, `r3batch` uses the `old copy` to rerun the job. For a description about the difference between the new and old copy of a rerunning job, refer to “Old copy and new copy of a rerunning job” on page 130.

To rerun a SAP Standard R/3 job from the Dynamic Workload Console, perform the following steps:

1. Click **Workload** > **Monitor Jobs**.
2. The list of defined tasks or queries for monitoring jobs is displayed. Select the hyperlink of a task to display the related list of jobs. If you have a predefined task to display SAP jobs, click that task.
3. In **Engine name**, select the name of the Tivoli Workload Scheduler engine connection from where you want to work with SAP jobs and click **OK**.
4. A list of jobs is displayed. Select an SAP Standard R/3 job.

   Distributed **Distributed environment**

   The **Job Type** column displays **r3** to help you identify SAP Standard R/3 jobs.

   z/OS **z/OS environment**

   The **Advanced Job Type** column displays **r3** to help you identify SAP Standard R/3 jobs.
5. Rerun the job.

   Distributed **Distributed environment**

   a. Click **Rerun**. The General properties for the rerun operation are displayed.
   b. Optionally, you can choose to not rerun the same job but instead, substitute the selected SAP job with a different job definition and run it. Type the job definition name in the **From Job Definition** field, or use the browse button to search for it and select it.
   c. Optionally, type the workstation name of the workstation on which you want to rerun the job in the **Workstation Name** field.
   d. Optionally, in **Step**, enter a specific numeric step of the SAP instruction from which you want to rerun the job rather than rerunning the whole job.
   e. Optionally, specify the start and finish time for the job.
   f. Click **Rerun**.

   The job reruns immediately or at the specified start time.

   z/OS **z/OS environment**

   In a z/OS environment, an alias for the job name is not required so the job reruns with the same name. The list of jobs always reports the latest action performed on the job.

   a. Before you can rerun a job, you must change the status of the job to **Ready**. Select a job and click **Set Status**.
   b. In **Change Status**, select **Ready**.
   c. Click **OK** to return to the list of jobs.

   The job reruns immediately and the internal status reports **Started**.

Old copy and new copy of a rerunning job

When the access method for SAP launches a job, it makes a copy of a template job and runs it.

The `new copy` feature is available for SAP versions 3.1i, and later. It copies an entire job, preserving steps, job class, and all print and archive parameters. It is performed by using a new SAP function module that is part of the SAP Support Package as stated in the SAP Notes 399449 and 430087.

The `old copy` feature, instead, is based on standard SAP function modules, and creates a new SAP job and adds the steps with a loop that starts from the step name or number you specified. Be aware that, unless you have XBP 2.0 or later:

- The `old copy` does not preserve all the print and archive parameters.
- The job class of the copy is always set to class `C`.

Refer to "Print parameter and job class issues" on page 89 to learn how to resolve the problem of lost job class and print and archive parameters.

SAP Note 758829 is required to ensure correct operation of the `new copy` and `old copy` features. See also Table 88 on page 286.

Defining SAP jobs dynamically

When you launch a job created as described in "Creating SAP Standard R/3 jobs from the Dynamic Workload Console" on page 106 and "Task string to define SAP jobs" on page 115, Tivoli Workload Scheduler makes a copy of the predefined job (also known as a template job) and runs the copy. If you want to run the job on several SAP systems, you must manually create the template job on each system.

This section describes how to create and submit SAP jobs dynamically without creating or referencing predefined job templates, by submitting:

- In the SAP system, a job that does not refer to an existing template in the SAP R/3 database.
- A job that references a predefined job template stored in the SAP R/3 database for which you want to change a parameter.

To take full advantage of this feature, make sure that you have XBP version 2.0 or later installed, because earlier versions of XBP do not support the full set of print and archive parameters, or provide a way to set the job class or the spool list recipient.

Task string to define SAP jobs dynamically

This section describes the task string that controls the running of SAP jobs. You can build an entire job definition by using the six main sections concerning SAP job parameters. These sections are grouped in Table 39 on page 134 and are related to the:

- Job
- Job steps
- Variants associated with the steps (for ABAP modules only)
- Spool list recipients associated with the job
- Flags associated with the job
- Tracing specifications for the job

You can specify them in the following places when you define their associated Tivoli Workload Scheduler jobs:

- In the **R/3 Command Line** section of the Task page of the Submit Ad Hoc Jobs action from the Dynamic Workload Console.
- In the **R3 Command Line** field of the More Options page of the SAP job definition, if you use the Dynamic Workload Console and selected a **SAP** job definition.
- As arguments of the `scriptname` keyword in the job definition statement, if you use the Tivoli Workload Scheduler command line.

- As arguments of the JOBCMD keyword in the JOBREC statement in the SCRIPTLIB of Tivoli Workload Scheduler for z/OS, if you are scheduling in an end-to-end environment. The following is an example of a JOBREC statement:

```
JOBREC
  JOBCMD('/-job job_name -user user_name -i job_ID -c class_value')
  JOBUSR(TWS_user_name)
```

To define and submit an SAP R/3 job dynamically, use the following syntax:

Job definition syntax

```
►►── -job──job_name──┬──────────────────┬── -flag type=exec──┬─────────────────────────┬──────────►
                     └─┬─ -i ──┬─job_ID─┘                    └─┬─ -host─┬─host_name─┘
                       └─ -id ─┘                               └─ -ts ──┘

►──┬─────────────────────┬──┬──────────────────────┬──┬─────────────────────────────┬────────────►
   └─ -sg──server_group─┘  └─ -client──source_client─┘  └─ -exec_client──target_client─┘

►──┬──────────────────────────────────┬──┬───────────────────┬─────────────────────────────────►
   └─ -rfc_client──rfc_logon_client─┘  └─ -c──class_value─┘

►──┬────────────────────────────────────────────┬──────────────────────────────────────────────►
   └─ -bdc_job_status_failed──bdc_processing─┘

►──┬───────────────────────────────────────────────────────────────────────────────────┬──────►
   └─┬─ -nobdc─────┬──┬─────────────────────────────────┬──┬───────────────────────────────┬─┘
     └─ -nobdcwait─┘  └─ -bapi_sync_level──┬─high───┬─┘  └─ -s──starting_step_number─┘
                                          ├─medium─┤
                                          └─low────┘

     ┌──────────────────────────────────────────────────────┐
     ▼                                                      │
►──┬──────────────────────────────────────────────────────┬─┴──────────────────────────────────►
   └─ -sStep_number─attribute_name─┬──────────────────┬─┘
                                   └─=attribute_value─┘

►──┬─────────────────────────────────────┬──┬─────────────────────────────────────────────┬────►
   └─ -vStep_number─variant_name─┘  └─ -vtxtStep_number─variant_description─┘

►──┬──────────────────────────────────────────┬───────────────────────────────────────────────►
   └─ -vparStep_number─name=variant_value─┘

►──┬───────────────────────────────────────────────────────────────────┬──┬────────────────────────┬──►
   └─ -vselStep_number─name=─┬─i─┬─#operation#lowest─┬──────────┬─┘  └─ -vtempStep_number─┘
                             └─e─┘                   └─#highest─┘

►──┬──────────────────────────────────┬──┬─────────────────────────────┬──┬────────────────────┬──►
   └─ -recipient──R/3_login_name─┘  └─ -rectype──recipient_type─┘  └─ -flag──┬─reccp─┬─┘
                                                                             └─recbl─┘
```

```
►─┬──────────────┬─┬──────────────┬─┬──────────────────┬──────────────────────────►
  └─ -flag recex─┘ └─ -flag recnf─┘ └─ -flag──┬─im────┬┘
                                              └─immed─┘

►─┬──────────────────────────────────┬─┬──────────────────────────────────┬────────►
  └─ -flag──┬─enable_applinfo──┬─────┘ └─ -flag──┬─enable_appl_rc──┬──────┘
            └─disable_applinfo─┘                 └─disable_appl_rc─┘

►─┬────────────────────────────────┬─┬───────────────────────────────────────────┬►
  └─ -flag──┬─enable_joblog──┬─────┘ └─ -flag──┬─enable_job_interceptable──┬─────┘
            └─disable_joblog─┘                 └─disable_job_interceptable─┘

►─┬───────────────────────────────────┬─┬─────────────────────┬─┬─────────┬────────►
  └─ -flag──┬─enable_spoollist──┬─────┘ └─ -flag──pc_launch───┘ └─ -debug─┘
            └─disable_spoollist─┘

►─┬───────────────────┬─┬────────────┬─┬────────────────────────────────┬─────────►◄
  └─ -tracelvl──┬─1─┬─┘ └─ -rfctrace─┘ └─ -rfc_client──rfc_logon_client─┘
                ├─2─┤
                └─3─┘
```

The following is an example of a definition for the SAPTEST job:

```
-job SAPTEST -C A -s1 program=BTCTEST -s1 type=A -s1 pr_release
-s2 report=BTCTEST -s2 variant=BVT –s2 type=A -flag type=exec
-vpar2 TESTNAME=test -vtxt2 Test
```

Table 39 on page 134 describes the parameters for the task string to define SAP jobs dynamically.

Note: The parameter values are case sensitive.

Table 39. Task string parameters for SAP jobs (dynamic definition)

Section	Parameters	Description
JOB	-job *job_name*	The name of the job to be run. This parameter is mandatory.
	-i *job_ID* -id *job_ID*	The unique SAP job ID. Specify this parameter if you are submitting a job that refers to a predefined job template stored in the SAP database for which you want to change a parameter.
	-host *host_name* -ts *host_name*	The name of the SAP workstation where the job is to be run. *host_name* has the format *hostname_SAPsystemname_SAPsystemnumber*. For example, the name of a host might be amss80a0_gs7_90 These parameters are mutually exclusive with -sg.
	-sg *server_group*	The name of the SAP server group where the job is to be run. Use this parameter to run the job on an application server that belongs to the group. The server group must exist on the SAP system, otherwise an error code is returned and the job is not launched. This parameter is case-sensitive and can be up to 20 characters. It is mutually exclusive with -host and -ts.
	-client *source_client*	The number that identifies the SAP client where the job definition is to be found, regardless of the client number defined by the **r3client** key in the options file. This parameter has no effect if a job ID is specified in the job definition.
	-exec_client *target_client*	The number that identifies the SAP client where the job is to be run, regardless of the client number defined by the `r3client` key in the options file. This parameter requires that the client-dependent data (such as the user name and report variants) exists on both the source and target clients. This parameter has no effect if a job ID is specified in the job definition.
	-rfc_client *rfc_logon_client*	The number that identifies the SAP client to be used for RFC logon. This value overwrites the value specified by the r3client keyword in the corresponding r3batch options file.
	-c *class_value*	The priority with which the job runs in the SAP system. Possible values are: **A** High priority **B** Medium priority **C** Low priority. This is the default value.
	-flag type=exec	Specify this parameter to enable the dynamic definition of the SAP job. This parameter is mandatory.

Table 39. Task string parameters for SAP jobs (dynamic definition) (continued)

Section	Parameters	Description
JOB	-bdc_job_status_failed *bdc_processing*	How Tivoli Workload Scheduler sets the completion status of a job running BDC sessions, according to a possible BDC processing failure. The allowed values are: *n* If at least *n* BDC sessions failed (where *n* is an integer greater than 0), Tivoli Workload Scheduler sets the job completion status as failed. **all** If all the BDC sessions failed, Tivoli Workload Scheduler sets the job completion status as failed. **ignore** When all the BDC sessions complete, regardless of their status, Tivoli Workload Scheduler sets the job completion status as successful. This is the default value. If **-nobdc** or **-nobdcwait** is set, this option is ignored.
	-nobdc -nobdcwait	Disables the BDC Wait option (enabled by default) to have the job considered as completed even if not all its BDC sessions have ended.
	-bapi_sync_level	Specifies the synchronization level between the SAP function modules BAPI_XBP_JOB_COPY and BAPI_XBP_JOB_START_ASAP. Allowed values are: **high** All RFC calls between BAPI_XBP_JOB_START_ASAP and BAPI_XBP_JOB_COPY are synchronized. This is the default. **medium** The RFC calls to BAPI_XBP_JOB_START_ASAP are synchronized. **low** The RFC calls are not synchronized.

Table 39. Task string parameters for SAP jobs (dynamic definition) (continued)

Section	Parameters	Description
STEP	-s *starting_step_number*	The number of the starting step.

	-s*step_number attribute_name=attribute_value*	The step number and its attributes, where:
		step_number
		The number of the step being defined. Each step is identified by a sequential number (1, 2, 3, ...*n*) using the *step number*.
		attribute_name
		The name of the attribute.
		attribute_value
		The value of the attribute. It is optional for some attributes.
		Attributes can be defined in any order, but cannot be repeated for the same step. Attribute validation is performed before the job is created in the SAP system. If the validation fails, the Tivoli Workload Scheduler job goes into the ABEND state. For a detailed description of each attribute and its values, see “Defining attributes for ABAP steps” on page 140 and “Defining attributes for external programs and external commands steps” on page 142.
		For example, the following step (step 8) is an ABAP module running the report "MYPGM" and has two attributes, only one of which has a value.
		`-s8 type=A -s8 program=MYPGM`
		`-s8 pr_cover="My title" -s8 pr_immed`

Table 39. Task string parameters for SAP jobs (dynamic definition) (continued)

Section	Parameters	Description
VARIANT[1]	-v*step_number* name	The variant name for the specified step number.
	-vtxt*step_number variant_description*	The textual description of the variant, in the Tivoli Workload Scheduler logon language (customizable with the `TWSXA_LANG` option of `r3batch`). The maximum length is 30 characters. Not valid for temporary variants.
	-vpar*step_number* name=*value*	For ABAP modules only. The value for a variant parameter for the specified step number. This parameter is mandatory when creating a new variant. For a complete list of the supported attributes for ABAP steps, see "Defining attributes for ABAP steps" on page 140.
	-vsel*step_number* name=*sign*#*operation*#*lowest*[#*highest*]	For ABAP modules only. The value for a variant selection option for the specified step number. **sign** Sign of the operation. Possible values are: **I** Include **E** Exclude **operation** Possible values are: **EQ** Equals **NE** Not equal to **BT** Between **NB** Not between **LT** Less than **LE** Less than or equal to **GT** Greater than **GE** Greater than or equal to **CP** Contains pattern **NP** Does not contain pattern **lowest** Low value of the selection. You can use up to 45 characters. **highest** High value of the selection. You can use up to 45 characters. This attribute is optional. For a complete list of the supported attributes for ABAP steps, see "Defining attributes for ABAP steps" on page 140.
	-vtemp*step_number*	For ABAP modules only. Specifies to assign a temporary variant to the specified step number. Temporary variants are created ad-hoc by the SAP system and assigned to the job instance when it is run. The lifecycle of the temporary variant is determined by the SAP system. If the job is deleted by SAP, then the temporary variant is deleted. See "Examples: Dynamically defining and updating SAP jobs" on page 144 to refer to examples that demonstrate the behavior of temporary variants.

Table 39. Task string parameters for SAP jobs (dynamic definition) (continued)

Section	Parameters	Description
SPOOL	-recipient *name*	The login name of an SAP user.
	-flag {reccp | recbl}	Specifies how the spool list is sent to the recipient. Possible values are: **reccp** The spool list is sent as a copy. **recbl** The spool list is sent as a blind copy.
	-flag recex	Specifies that the spool list is sent as an express message to the recipient.
	-flag recnf	Specifies that the recipient is not allowed to forward the spool list.
	-rectype *type*	Specifies the recipient type. Possible values are: **blank** SAP user (default value) **B** SAP user **C** Shared distribution list **D** X.500 address **G** Organization object/ID **H** Organization unit **I** SAP object **L** Telex number **O** SAPoffice user **P** Private distribution list **R** SAP user in another SAP system **U** Internet address **1** Other recipient type
FLAGS	-flag im -flag immed	Specifies to launch the job immediately, meaning that if there are no spare work processes, the job fails.
	-flag enable_applinfo -flag disable_applinfo	Enables or disables the retrieval and appending of the SAP application log to the stdlist of Tivoli Workload Scheduler.
	-flag enable_appl_rc -flag disable_appl_rc	Enables or disables the mapping of the SAP application return code to the Tivoli Workload Scheduler return code.
	-flag enable_joblog -flag disable_joblog	Enables or disables retrieval of the joblog.
	-flag enable_joblog -flag disable_joblog	Enables or disables retrieval of the joblog.
	-flag enable_job_interceptable -flag disable_job_interceptable	Enables or disables the job launched by r3batch to be intercepted by SAP. If enabled, when r3batch launches a job and the SAP job interception feature is enabled, the job can be intercepted if it matches previously defined criteria. If disabled, the job launched by r3batch cannot be intercepted by SAP. This setting overwrites the setting in the common options file.

Table 39. Task string parameters for SAP jobs (dynamic definition) (continued)

Section	Parameters	Description
-flag pc_launch	Specifies to launch child jobs that are in scheduled state. **ON** The product launches child jobs that are in scheduled state. **OFF** The product does not launch child jobs that are in scheduled state. This is the default value. **Note:** You can use this option only if you activated the parent-child feature on the SAP system. On the XBP 2.0 (or later) SAP system, you activate this feature by using the INITXBP2 ABAP report.	
TRACING	-debug	Enables maximum trace level.
	-tracelvl 1|2|3	Specifies the trace setting for the job. Possible values are: **1** Only error messages are written in the trace file. This is the default. **2** Informational messages and warnings are also written in the trace file. **3** A most verbose debug output is written in the trace file. For more details, refer to "Configuring the tracing utility" on page 281.
	-rfctrace -trace	Enables RFC trace.

Note: See "Examples: Dynamically defining and updating SAP jobs" on page 144 to refer to examples that demonstrate the behavior of variants and temporary variants.

1. The following rules apply when you create or update SAP jobs dynamically:
   - To create or reference a variant within an ABAP step, you can use one of the following equivalent syntaxes:
     - -s1 Variant=Var1
     - -s1 Parameter=Var1
     - -v1 Var1
   - If a variant does not exist, it is created with the parameters specified in the job definition statement. In this case, all the required attributes of the variant must be given a value. You cannot create empty variants. For example, if you specify **`-vtemp1`**, with no value assigned, an empty temporary variant is erroneously created.
   - If a variant is already present in the SAP system, its values are modified according to the command line parameters. If the existing variant is an extended one, a new instance of it is created with resolved placeholders and updated counters. This new variant instance is then updated using the values from the command line. Finally, the job step is run using this variant instance.
   - All changes to the variant values are permanent. That is, Tivoli Workload Scheduler neither restores the old values of the variants, nor deletes the variants created after the job is run. Tivoli Workload Scheduler does not change the case of the variant values.

Defining attributes for ABAP steps

To create and submit SAP jobs dynamically, look at the table and define the attributes for ABAP steps.

Table 40 shows a complete list of the supported attributes for ABAP step module definition:

Table 40. Supported attributes for ABAP step definition

Attribute name	Synonym	Description	Required
type	typ	Specify the step type. Possible values are: • A • ABAP The product performs a check for correct attribute values prior to launching the job.	✓
program		Specify the ABAP program name.	✓
parameter		Specify the ABAP variant name.	✓
user	authcknam	Specify the user of the step.	✓
language	lang	Specify the step language. This attribute accepts language names in either the ISO format (two characters, for example DE, EN) or the R/3 format (one character, for example D, E). If this attribute is not specified, the login language of the access method is used (customize using the option **twsxa_lang** in the r3batch options files). The product performs a check for a valid language prior to launching the job.	✓
pr_dest	printer pdest	Print Parameter: Specify the printer for the output.	
pr_copies	prcop	Print Parameter: Specify the number of copies. The value of this attribute must be numeric. A corresponding check is performed prior to launching the job.	
pr_lines	linct	Print Parameter: Specify the page length. The value of this attribute must be numeric. A corresponding check is performed prior to launching the job.	
pr_columns	linsz	Print Parameter: Specify the page width. The value of this attribute must be numeric. A corresponding check is performed prior to launching the job.	
pr_auth	prber	Print Parameter: Authorization	
pr_arcmode	armod	Print Parameter: Archiving mode	
pr_sapbanner	prsap	Print Parameter: SAP cover page	

Table 40. Supported attributes for ABAP step definition (continued)

Attribute name	Synonym	Description	Required
pr_exp	pexpi	Print Parameter: Spool retention period The value of this attribute must be a single digit. A corresponding check is performed prior to launching the job.	
pr_recip	prrec	Print Parameter: Recipient	
pr_spoolname	plist	Print Parameter: Name of spool request [1]	
pr_format	paart	Print Parameter: Print formatting [1]	
pr_dep	prabt	Print Parameter: Department on cover page [1]	
pr_spoolds	prdsn	Print Parameter: Name of spool data set [1]	
pr_spoolprio	priot	Print Parameter: Spool request priority [1]	
pr_immed	primm	Print Parameter: Print immediately [2]	
pr_release	prrel	Print Parameter: Delete after printing [2]	
pr_banner	prbig	Print Parameter: Selection cover page [2]	
pr_newspool	prnew	Print Parameter: New spool request [1] [2]	
pr_cover	prtxt	Print Parameter: Text for cover page [1]. If the string contains spaces it must be enclosed between single quotes.	
pr_hostcover	prunx	Print Parameter: Host spool cover page [1] . Possible values are: **' '** Blank. Does not use any cover page. **'X'** Prints the host cover page. **'D'** Prints the default host cover page.	
al_sapobject	sap_object	SAP ArchiveLink: Object type of business object	
al_object	object	SAP ArchiveLink: Document type	
al_info	info	SAP ArchiveLink: Info field	
al_id	archiv_id	SAP ArchiveLink: Target storage system [1]	
al_doctype	doc_type	SAP ArchiveLink: Document class [1]	
al_rpchost	rpc_host	SAP ArchiveLink: PRC host [1]	
al_rpcserv	rpc_servic	SAP ArchiveLink: RPC service / RFC destination [1]	
al_iface	interface	SAP ArchiveLink: Name of communication connection component [1]	
al_client	mandant	SAP ArchiveLink: Client [1]	
al_report		SAP ArchiveLink: Report name [1]	
al_text	arctext	SAP ArchiveLink: Text information field [1]	
al_date	datum	SAP ArchiveLink: Archiving date [1]	
al_user	arcuser	SAP ArchiveLink: Data element for user [1]	
al_printer		SAP ArchiveLink: Target printer [1]	
al_format	formular	SAP ArchiveLink: Output format [1]	
al_path	archivpath	SAP ArchiveLink: Standard archive path [1]	
al_protocol	protokoll	SAP ArchiveLink: Storage connection protocol [1]	

Table 40. Supported attributes for ABAP step definition (continued)

Attribute name	Synonym	Description	Required
al_version		SAP ArchiveLink: Version number [1]	

Note:

1. This attribute is available for BC-XBP 2.0 and later.
2. This attribute is a flag, that is, it does not have a value, for example: `–s2 pr_release`.

Tivoli Workload Scheduler performs the following syntax validation on job attributes:

- Only valid attributes are allowed.
- Checks if a particular attribute requires a value.
- The values of the following attributes are checked:
  - type
  - language
  - pr_copies
  - pr_lines
  - pr_colums

Validation is performed before the job is created in the SAP system. If the validation fails, the Tivoli Workload Scheduler job goes into the ABEND state.

Defining attributes for external programs and external commands steps

Table 41 shows a complete list of the supported attributes for external programs and external commands step definition.

Table 41. Supported attributes for external programs and external commands step definition

Attribute name	Synonym	Description	Required
type	typ	The step type can assume one of the following values: **For external programs** • X • EXTPRG **For external commands** • C • EXTCMD Before launching the job, the product performs a check for correct attribute values.	✓
report		ABAP program name or name of the external program or command.	✓
parameter		Parameters for the external program or command.	
user	authcknam	User of the step.	

Table 41. Supported attributes for external programs and external commands step definition (continued)

Attribute name	Synonym	Description	Required
language	lang	Step language. This attribute accepts language names in either the ISO format (two characters, for example DE, EN) or the R/3 format (one character, for example D, E). If this attribute is not specified, the login language of the access method is used (customize using the **twsxa_lang** option in the `r3batch` option files). The product performs a check for a valid language prior to launching the job.	
targethost	xpgtgtsys	Target host for the external program or command. This name must be exactly the same as the name shown in the External Operating System Commands table in the SAP system (transaction `sm69`).	
os	opsystem	Operating system for the external command. This name must be exactly the same as the name shown in the External Operating System Commands table in the SAP system (transaction `sm69`).	
termcntl	waitforterm	Control flag: if an external command or program is to be run synchronously. [2]	
tracecntl		Control flag: if SAP tracing level 3 is activated for tracing SAPXPG, the program that starts an external command or program. [1] [2]	
stdoutcntl		Control flag: indicates if standard output from an external command or program is to be written to the job log. [1] [2]	
stderrcntl		Control flag: indicates if standard error from an external command or program is to be written to the job log. [1] [2]	

Note:

1. This attribute is available for BC-XBP 2.0 and later.
2. This attribute is a flag, that is, it does not have a value, for example: `–s2 pr_release`.

Tivoli Workload Scheduler performs the following syntax validation on job attributes:

- Only valid attributes are allowed.
- Checks if a particular attribute requires a value.
- The values of the following attributes are checked:
  - type
  - language
  - pr_copies

– pr_lines
– pr_colums

Validation is performed before the job is created in the SAP system. If the validation fails, the Tivoli Workload Scheduler job goes into the ABEND state.

Specifying job parameters using variable substitution

Parameters can be provided at run time using the variable substitution feature. For example, the value appears as:

```
-s1 report=&VARNAME
```

The variable substitution process occurs while Tivoli Workload Scheduler is creating the symphony file.

Examples: Dynamically defining and updating SAP jobs

This section describes some usage examples of this feature:

Job definition and run scenario using the -flag type=exec parameter

The following example creates and runs a 3-step job. The first step runs the ABAP MYPROG1 using variant VAR01 and associated variant parameter. Step 2 has a step user defined. Step 3 uses the same ABAP as step 1 with no associated variant.

The only requirement is that the elements referred to are known in the SAP system (user, program). If the variant does not exist, there should be a set of values to define the content of the variant for its creation (pairs of `–vpar`*N* `-vsel`*N* parameters for the parameters and selections of the ABAP program).

```
-job TESTJOB01 –c A
-s1 type=A –s1 program=MYPROG1
–v1 VAR01 –vpar1 TESTNAME=TST
-s2 report=SPOOLX1 –s2 user=PRTUSER
–s3 type=A –s3 program=MYPROG1 –flag type=exec
```

The job returns job ID 12345678

Job copy and overwrite the job created in the previous step

The following job statement references the job created in the previous example. A new copy of the job is made and the parameters specified in the invocation are used to update the definition. In this case the variant for step 1 is modified and a new external program step (Step 4) is added.

```
-job TESTJOB01 –i 12345678
–s1 variant=VAR01A
–vpar1 TESTNAME=TST2
–s4 type=X -s4 report=niping –s4 parameter=-t
-flag type=exec
```

Copy and overwrite a job referencing an existing job template

The following example shows a job creation referencing a job template (previously created without using this feature). A template job called TEMPLAJOB already exists on the SAP system with an ID of 56780123. It is a single ABAP step job to which we now add some print parameters.

```
-job TEMPLAJOB
–I 56780123 -s1 pr_immed
-flag type=exec
```

A temporary variant is created using the information indicated in the expression

The following is the syntax to be used:

```
-vpar1 <parameter_name>=<parameter_value> ...
-vsel1 <selection_option_name>
... -vtemp1
```

The following example shows how you can submit a job that creates a temporary variant that is assigned to step number 1, and assigns a value to a variant parameter for step number 1:

```
-job TESTJOB01 -C A -flag type=exec -user R3USER
-s1 type=A -s1 program=MYPROG1
-vtemp1 -vpar1 TESTNAME=TST
```

The following example shows how you can submit a job that creates a temporary variant that is assigned to step number 1, assigns a value to a variant parameter for step number 1, and assigns a value to a variant selection option (date) for step number 1:

```
-job TESTJOB01 -C A -flag type=exec -user R3USER
-s1 type=A -s1 program=MYPROG1
-vtemp1 -vpar1 FILENAME=FLN
-vsel1 date=E#BT#20110101#20110412
```

Assign a temporary variant to the specified step number

The following is the syntax to be used:

```
-v1 <temporary_variant_name> -vtemp1
```

The following is an example of how you can submit a job to assign a temporary variant, which has already been created (as in the previous example), and assign a value to step number 1:

```
-job TESTJOB01 -C A -flag type=exec -user R3USER
-s1 type=A -s1 program=MYPROG1
-vtemp1 -v1 &000000000001
```

The value for a temporary variant that already exists is substituted with the value indicated in the expression

The following is the syntax to be used:

```
-v1 <temporary_variant_name> -vpar1 <parameter_name>=<parameter_value> ...
-vsel1 <selection_option_name> ... -vtemp1
```

The following is an example of how you can submit a job that substitutes the value of a temporary variant, which must already exist, with a new value. The temporary variant must exist, otherwise, the expression returns an error.

```
-job TESTJOB01 -C A -flag type=exec -user R3USER
-s1 type=A -s1 program=MYPROG1
-vtemp1 -v1 &000000000001 -vpar1 TESTNAME=TST2
```

Defining conditions and criteria

Tivoli Workload Scheduler for Applications accesses the Computer Center Management System (CCMS) Background Processing components of SAP systems through the BC-XBP interface to provide additional capabilities from the Dynamic Workload Console, one of those being the Criteria Manager.

Tivoli Workload Scheduler for Applications supports the BC-XBP 3.0 interface which provides functions to control R/3 batch jobs.

The Criteria Manager is a tool that enables you to define conditions and criteria that, when combined, form complex dependencies that you can use in the following contexts:

- Managing raised events in the SAP event history.
- Managing reorganization tasks against the SAP event history.
- Intercepting jobs.

If you have other types of criteria defined on your SAP system, then you can perform other actions in addition to those listed in this section.

The criteria profile

The Criteria Manager enables you to define a criteria profile which is a container for a combination of criteria. The criteria profile can be of various types and each criteria type has a standard set of selection criterion. For each criteria, you can specify a single value, a range of values by indicating a lower and upper limit, and multiple values. The following is the standard set of selection criterion for each criteria profile type. In addition to these, you can also see any other types of criteria profiles you have defined on your SAP system:

Event History

EVENTID
The identifier of the event defined in the SAP system.

EVENTPARM
The parameter of the event defined in the SAP system.

PARMID
The identifier of the parameter of the event defined in the SAP system.

Event History Reorg

Event State
The state of the event.

Event Timestamp
The timestamp for the event.

Interception

Job Name
A name identifying the job.

Job Class
The class assigned to the job that represents the priority with which the job runs in the SAP system.

The criteria hierarchy

You create and combine criteria in a criteria hierarchy. The criteria hierarchy is a set of all the criteria that must be fulfilled for a specific action to take place in the specific context. For example, you can define a criteria hierarchy to log all raised events in the SAP event history with an event name that begins with "CRITICAL_EVENT" and with an event argument equal to 150.

The criteria in the hierarchy is grouped in nodes and relationships between the nodes are determined by the logical operators AND or OR. You can nest nodes in other nodes.

To have the criteria profile begin processing, the criteria profile must be activated. Only one criteria profile of the same type can be active at one time.

An example

See "Example: Defining which raised events to log" for an example that demonstrates how to build a criteria hierarchy to manage the logging of raised events in the SAP event history.

Example: Defining which raised events to log

The event history stores all events that are raised by the system. You can define specific criteria so that only raised events that match certain criteria are logged.

The event history enables Tivoli Workload Scheduler for Applications to consume events that are raised by the SAP system.

Checking the log of raised events gives you access to the following information:

- Verify that an event was raised in the system.
- Verify if the event was processed.

In the example that follows, an event history criteria profile is created that contains the definition of the criteria, the criteria hierarchy, that events must fulfill to be logged in the event history. The criteria profile must then be activated so that it can begin processing events according to the criteria.

The criteria profile, `Event profile 1`, contains a criteria hierarchy that logs only those events in the event history with event name that begins with `CRITICAL_EVENT` and event argument equal to "789".

Create the criteria profile

A criteria profile contains the definition of the criteria you want to set for logging raised events.

Create a criteria profile, `Event profile 1`, of type, Event History, to contain the criteria hierarchy.

1. In the portfolio on the left, click **Tivoli Workload Scheduler** > **Workload** > **Design** > **Manage SAP Criteria Profiles**.
2. In **Engine name**, select the name of the Tivoli Workload Scheduler engine connection from which you want to work with SAP jobs.
3. In **Workstation name**, type the name of the workstation where the SAP job runs. This is the workstation with the r3batch access method that communicates with the remote SAP system. If you do not know the name of the workstation, click the **Lookup Workstations** icon to enter your filter criteria and click **Search**. If you enter a string representing part of the workstation name, it must be followed by the asterisk (*) wildcard character. Both the question mark (?) and asterisk (*) are supported as wildcards. You can also simply use the asterisk wildcard character (*) to display all workstations. Optionally, specify any of the other search criteria available and click **Search**. From the results displayed, select the workstation and click **OK**.
4. In **Options file**, specify an options file that resides on the specified workstation. Each workstation can have one or more options files that can be used to customize the behavior of the r3batch access method, except for extended agent workstations, where only one options file can exist and therefore does not need

to be specified. For the workstation specified, enter the file name of the options file or click pick icon to search for options files that reside on the specified workstation and select one.

5. Click **Go**.
6. From the Criteria Manager main view, click **New** to create a criteria profile.
7. Select **Event History** as the type of criteria profile you want to create.
8. Enter descriptive text that enables you to easily identify the criteria profile in the table of criteria profiles. Type `Event profile 1`. Avoid using special characters such as, < (less than), > (greater than), or the ' (apostrophe) in this field.
9. Click **Save**.

The criteria profile is displayed in the list of criteria profiles and it is not yet active.

Next, begin building the criteria hierarchy. The criteria profile is the container for the criteria hierarchy.

Build the criteria hierarchy

The criteria hierarchy is stored in the criteria profile and is made up of criteria. A group of criteria is contained in a node.

The criteria hierarchy is made up of a combination of nodes and criteria. A node contains a group of criteria where the relationship between the criteria is determined by an AND or an OR relation. You can nest nodes in other nodes. By default, a top level AND node is created in the criteria hierarchy. You can create other AND or OR nodes nested in this parent node. You can also add one or more criterion to the nodes. Add an AND node when all of the criteria defined in the node must be fulfilled. Add an OR node when at least one of the criteria defined in the node must be fulfilled.

In this example, define a criterion that logs all events whose name begins with `CRITICAL_EVENT` and with event argument equal to `789`.

1. Click 🗋 to create a new criterion in the default AND node.
2. In **Description**, type `Criterion 1`.
3. In **EVENTID**, click ➕ to specify the value for the EVENTID field.
4. Leave the default value **Select** to indicate to use the selection criterion specified when processing events.
5. In **Options**, select **Pattern** and in **Single Value or Lower Limit**, type `CRITICAL_EVENT*`. This sets the condition for the event name.
6. In **EVENTPARM**, click ➕ to specify the value for the EVENTPARM field.
7. Leave the default value **Select** to indicate to use the selection criterion specified when processing events.
8. In **Options**, select **Equal to** and in **Single Value or Lower Limit**, type `789`. This sets the condition for the event argument.
9. Click **Save** to save the criterion definition.

The criteria profile now contains a criterion that specifies which raised events must be logged. You can continue to create another criteria in the same parent node or you can nest either an AND or an OR node in the parent node to determine the logical relation between the criteria that the nested node will contain. Add an AND node within which you can create one or more criteria where all the criteria specified in the node must be fulfilled, or add an OR node within which you can create one or more criteria where at least one of the criteria specified must be fulfilled.

To apply this criteria profile so that it begins processing events according to the criteria defined, you must activate the criteria profile.

Activate the criteria profile

To apply the `Event profile 1` criteria profile so that it begins processing raised events according to the criteria specified in the criteria hierarchy, you must activate the criteria profile.

A criteria profile can either be active or not active. For a criteria profile to take effect, the profile must be activated. Only one criteria profile of the same type can be active at one time. Criteria profiles cannot be edited if they are in the active state. Follow the procedure to activate the `Event profile 1` criteria profile.

1. Select the `Event profile 1` criteria profile from the table of criteria profiles.
2. Select **Activate** from the toolbar.✓

The status of the criteria profile is updated to show that it is now active. The criteria profile can now begin to process raised events according to the specifications of the criteria hierarchy and log them to the event history. If another criteria profile of the same criteria type was active, its status changes to inactive.

Using the BDC Wait option

With the Batch Data Collector (BDC) Wait option, you can specify that an R/3 job launched by Tivoli Workload Scheduler is not to be considered complete until all of its BDC sessions have completed. This prevents other Tivoli Workload Scheduler jobs that are dependent on the R/3 job from being launched until all of the related BDC sessions for the R/3 job have ended.

To make use of the option, an R/3 job must write informational messages in its job log. This can be done by modifying the SAP function module BDC_OPEN_GROUP as follows:

```
FUNCTION BDC_OPEN_GROUP.
...
CALL 'BDC_OPEN_GROUP' ID 'CLIENT'      FIELD CLIENT
                      ID 'GROUP'       FIELD GROUP
                      ID 'USER'        FIELD USER
                      ID 'KEEP'        FIELD KEEP
                      ID 'HOLDDATE'    FIELD HOLDDATE
                 ID 'DESTINATION' FIELD DEST
                 ID 'QID'         FIELD QID
                 ID 'RECORD'      FIELD RECORD
                 ID 'PROG'        FIELD PROG.
*
IF SY-SUBRC EQ 0.
  BQID   = QID.
  BUSER  = SY-MSGV1.
  BGROUP = GROUP.
* CALL FUNCTION 'DB_COMMIT'.
  CALL FUNCTION 'ENQUEUE_BDC_QID'
  EXPORTING   DATATYP   = 'BDC '
  GROUPID    = BGROUP
  QID              = BQID
  EXCEPTIONS  FOREIGN_LOCK  = 98
  SYSTEM_FAILURE   = 99.

  IF SY-SUBRC EQ 0.
     message i368(00) with 'BDCWAIT: ' qid.
  ENDIF.

ENDIF.
*
PERFORM FEHLER_BEHANDLUNG USING SY-SUBRC.
*
*
ENDFUNCTION.
```

Note: The actual parameters of the call of the C function (`CALL 'BDC_OPEN_GROUP' ID ...`) might vary depending on the SAP release. With this approach, you obtain a global change in your R/3 system.

The completion status of an R/3 job launched by Tivoli Workload Scheduler is based on the value you set for the **bdc_job_status_failed** option. By default, this option is set to `ignore`, meaning that the job is considered successfully completed when the BDC sessions are finished, regardless of their success or failure. For details about the **bdc_job_status_failed** option, refer to Table 37 on page 118.

Job interception and parent-child features

This section describes how the job interception and parent-child features of BC-XBP 2.0 and 3.0 are supported by Tivoli Workload Scheduler for Applications.

Note: Distributed The process of defining relaunch criteria and collecting and relaunching intercepted jobs is supported only in distributed environments and not in z/OS environments.

Implementing job interception

The high-level steps required to implement job interception.

Job interception is a feature of both the BC-XBP 2.0 and BC-XBP 3.0 interfaces. It enables Tivoli Workload Scheduler to have a very sophisticated control over the jobs launched by SAP R/3 users from the SAP graphical interface.

The job interception mechanism becomes active when the SAP R/3 job scheduler is about to start an SAP R/3 job (that is, when the `start` conditions of an SAP R/3 job are fulfilled). It checks the job parameters (`job name`, `creator`, `client`) against the entries in the SAP R/3 table `TBCICPT1`, and when the job parameters match the criteria, the SAP R/3 job is set back to the `scheduled` status and is marked with a special flag, denoting that the job has been intercepted. The criteria defined in the criteria table establishes which job are intercepted.

If Tivoli Workload Scheduler has been set up to handle job interception, it periodically runs its own job to retrieve a list of intercepted jobs and reschedules them to be relaunched. This job can be referred to as the *interception collector* job.

Job interception with the BC-XBP 2.0 interface is based on the single extended agent workstation, whereas with the BC-XBP 3.0 interface, job interception is based on the currently active job interception criteria profile.

Note:

- Jobs launched by Tivoli Workload Scheduler, or by any other external scheduler using the BC-XBP interface, can be intercepted provided the **job_interceptable** option in the common options file is set to ON, and the **-flag enable_job_interceptable** keyword is included in the job definition.
- Ensure that the job interception and job throttling features are not running at the same time. The interception collector jobs fail if a job throttler instance is running. To stop the job throttler, refer to "Step 5. Starting and stopping the job throttling feature" on page 176.

The following are the high-level steps required to implement job interception for both the BC-XBP 2.0 and 3.0 interfaces.

Job interception and the BC-XBP 2.0 interface

To set up Tivoli Workload Scheduler to handle job interception in an SAP R/3 environment with the BC-XBP 2.0 interface, implement the following steps:

1. Install the BC-XBP 2.0 interface. Refer to SAP Note 604496 to know if your SAP R/3 system already has the BC-XBP 2.0 interface, or which SAP R/3 support package you need to install to enable it.
2. Define a Tivoli Workload Scheduler job to periodically collect the intercepted SAP R/3 jobs.
3. Specify interception criteria in the SAP R/3 system.
4. Specify interception criteria in Tivoli Workload Scheduler from the Monitor Workstations portlet on the Dynamic Workload Console. The criteria is set at workstation level.
5. Activate the job interception feature of the BC-XBP 2.0 interface.

Job interception and the BC-XBP 3.0 interface

To set up Tivoli Workload Scheduler to handle job interception in an SAP R/3 environment with the BC-XBP 3.0 interface, implement the following steps:

1. Verify if the BC-XBP 3.0 interface is already installed on the SAP R/3 system.
2. Define a Tivoli Workload Scheduler job to periodically collect the intercepted SAP R/3 jobs.
3. Specify interception criteria in the SAP R/3 system.
4. Specify interception criteria in Tivoli Workload Scheduler from the Manage SAP Criteria Profiles portlet on the Dynamic Workload Console.
5. Activate the job interception feature of the BC-XBP 3.0 interface.

Collecting intercepted jobs periodically for BC-XBP 2.0

With the BC-XBP 2.0 interface, you can configure the job interception collector using a Tivoli Workload Scheduler job that periodically retrieves intercepted jobs and relaunches them.

Define a Tivoli Workload Scheduler job that uses the SAP R/3 interception collector task to collect intercepted jobs and restart them.

To define a Tivoli Workload Scheduler job that collects intercepted job and relaunches them, use the following syntax:

```
XANAME#JOBNAME
  SCRIPTNAME "TWS_home/methods/r3batch -t HIJ -c XANAME"
  DESCRIPTION "Collects intercepted jobs on SAP XA XANAME"
  STREAMLOGON TWSuser
  RECOVERY STOP
```

Where:

XANAME
Name of the extended agent workstation.

JOBNAME
Name of the Tivoli Workload Scheduler job.

TWS_home
Fully qualified path to your Tivoli Workload Scheduler installation.

–t HIJ This is the SAP R/3 task type to run the job interception collector. HIJ stands for `Handle Intercepted Jobs`.

TWSuser
Name of the Tivoli Workload Scheduler user that launches the access method.

The interception collector job runs at periodical intervals; for example, every 10 minutes. It retrieves all the jobs that have been intercepted since the last run of the interception collector, and launches them again according to a template.

Collecting intercepted jobs periodically for BC-XBP 3.0

With the BC-XBP 3.0 interface, you can configure the job interception collector using a Tivoli Workload Scheduler job that periodically retrieves intercepted jobs and relaunches them.

Because intercepted jobs remain in the `Released` and then `Intercepted` status until they are relaunched, you need to use the SAP R/3 interception collector task to collect and relaunch them.

To define a Tivoli Workload Scheduler job that collects and relaunches jobs use the following syntax:

```
ENGINE_NAME_HOSTING_XA#JOBNAME
DOCOMMAND "TWS_home/methods/r3batch -t HIJ -c ENGINE_NAME_HOSTING_XA -- \
   "-profile_id <profile_ID_number>\""
STREAMLOGON TWSuser
DESCRIPTION "Collects intercepted jobs on SAP ENGINE_NAME_HOSTING_XA"
TASKTYPE UNIX
RECOVERY STOP
```

where,

ENGINE_NAME_HOSTING_XA
The name of the engine workstation hosting the XA workstation with the r3batch access method that communicates with the SAP system.

JOBNAME
Name of the Tivoli Workload Scheduler job.

TWS_home
Fully qualified path to your Tivoli Workload Scheduler installation.

–t HIJ This is the SAP R/3 task type to run the job interception collector. HIJ stands for `Handle Intercepted Jobs`.

- profile_id <*profile_ID_number***>**
Specifies the identification number of the interception criteria profile on the SAP system for XBP 3.0.

TWSuser
Name of the Tivoli Workload Scheduler user that launches the access method.

The interception collector job runs at periodical intervals; for example, every 10 minutes. It retrieves all the jobs that have been intercepted since the last run of the interception collector, and launches them again according to a template.

Note: If the interception collector is configured for XBP 3.0 job interception, but the XBP 2.0 interface is configured on the SAP system, the collector fails. Ensure the XBP interface versions are synchronized.

Setting interception criteria on the SAP R/3 system

In SAP R/3, the interception criteria are held in table `TBCICPT1`. Only jobs that match the criteria of this table are intercepted, when their `start` conditions are fulfilled. All the other jobs are run normally.

You can maintain the entries in this table by using transaction `se16` and setting the following:

- Client number
- Job mask
- User mask

Setting interception criteria on Tivoli Workload Scheduler

In Tivoli Workload Scheduler, interception criteria are defined and used by setting:

Table criteria

For BC-XBP 2.0

You set table criteria from the **Monitor Workstations** portlet from the Dynamic Workload Console.

For details about how you set table criteria, see “Setting SAP R/3 table criteria on the extended agent workstation.”

For BC-XBP 3.0

You set table criteria from the **Manage SAP Criteria Profiles** portlet from the Dynamic Workload Console.

For details about how you set table criteria, see “Setting SAP R/3 criteria in the job interception criteria profile” on page 154.

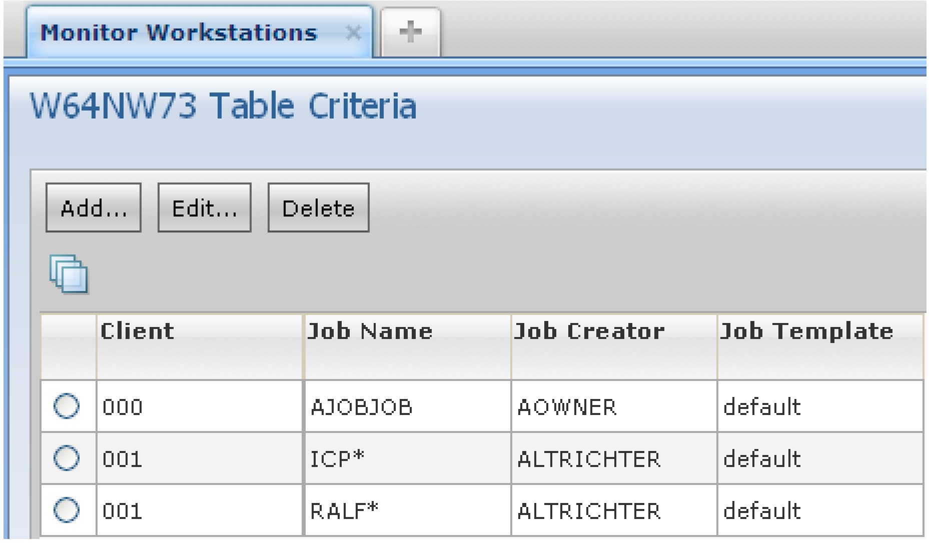
Template files (optional)

For details about how you create template files, see “Using template files” on page 155.

Setting SAP R/3 table criteria on the extended agent workstation:

To set table criteria with the BC-XBP 2.0 interface on an SAP R/3 job using the **Monitor Workstations** portlet on the Dynamic Workload Console, perform the following steps:

1. In the portfolio on the left, click **Tivoli Workload Scheduler** > **Scheduling Environment** > **Monitor** > **Monitor Workstations**.
2. Click the task for the workstations you want to monitor.
3. Select an engine and click **OK**.
4. Select an extended agent workstation in the table of displayed workstations, and click **More Actions** > **Table Criteria** from the toolbar.
5. The Table Criteria panel displays. From this panel you can add, delete, edit, or refresh criteria.



	Client	Job Name	Job Creator	Job Template
○	000	AJOBJOB	AOWNER	default
○	001	ICP*	ALTRICHTER	default
○	001	RALF*	ALTRICHTER	default

Figure 8. The Table Criteria panel

6. Specify the criteria:
   a. In **Client**, specify the client workstation of the SAP R/3 job.
   b. In **Job Name**, specify a filter to match a set of SAP R/3 jobs. Use the asterisk (*) wildcard character to match a set of jobs.
   c. In **Job Creator**, specify a filter to match a set of SAP R/3 job creator. Use the asterisk (*) wildcard character to match a set of jobs.
   d. Optionally, in **Job Template**, specify the template file that contains instructions for the interception collector about how to run the intercepted SAP R/3 job under control of Tivoli Workload Scheduler. For more information about template files, see "Using template files" on page 155.
   e. In **Job Class**, specify the class assigned to the job that represents the priority with which the job runs on the SAP system.
7. Click **OK**.

Setting SAP R/3 criteria in the job interception criteria profile:

Setting criteria to intercept jobs and relaunch them.

To set the criteria that defines which SAP R/3 jobs to intercept and relaunch with the BC-XBP 3.0 interface using the Dynamic Workload Console, perform the following steps:

1. In the portfolio on the left, click **Tivoli Workload Scheduler** > **Workload** > **Design** > **Manage SAP Criteria Profiles**.
2. In **Workstation name**, type the name of the workstation where the SAP job runs. This is the workstation with the r3batch access method that communicates with the remote SAP system. If you do not know the name of the workstation, click the **Lookup Workstations** icon to enter your filter criteria and click **Search**. If you enter a string representing part of the workstation name, it must be followed by the asterisk (*) wildcard character. Both the question mark (?) and asterisk (*) are supported as wildcards. You can also simply use the asterisk wildcard character (*) to display all workstations. Optionally, specify any of the other search criteria available and click **Search**. From the results displayed, select the workstation and click **OK**.
3. From the Criteria Manager main view, click **New** to create a criteria profile.

4. Select **Interception** as the type of criteria profile you want to create.
5. Enter descriptive text that enables you to easily identify the criteria profile in the table of criteria profiles. Avoid using special characters such as, < (less than), > (greater than), or the ' (apostrophe) in this field.
6. Click **Save**. The criteria profile is displayed in the list of criteria profiles and it is not yet active.
7. On the **Details** tab in the upper-right pane, define the criteria that intercepted jobs must match. For example, to intercept jobs with a job name beginning with "ICP", specify the following criteria:
   a. Click 🗋 to define a new criterion.
   b. In **Description**, type `Criterion 1`.
   c. In **JOB NAME**, click ➕ to specify the value for the JOB NAME field.
   d. Leave the default value **Select** to indicate to use the selection criterion specified when intercepting jobs.
   e. In **Options**, select **Pattern** and in **Single Value or Lower Limit**, type `ICP*`. This sets the condition for the job name.
   f. Click **Save** to save the criterion definition.
8. Define the criteria that must be matched to relaunch intercepted jobs. Click the **Job Relaunch Criteria** tab.
   a. Click 🗋 to define a new criteria that determines which jobs are relaunched.
   b. In **Client**, specify the client workstation of the SAP R/3 job.
   c. In **Job Name**, specify a filter to match a set of SAP R/3 jobs. Use the asterisk (*) wildcard character to match a set of jobs.
   d. In **Job Creator**, specify a filter to match a set of SAP R/3 job creator. Use the asterisk (*) wildcard character to match a set of jobs.
   e. Optionally, in **Job Template**, specify the template file that contains instructions for the interception collector about how to run the intercepted SAP R/3 job under control of Tivoli Workload Scheduler. For more information about template files, see "Using template files."
   f. In **Job Class**, specify the class assigned to the job that represents the priority with which the job runs on the SAP system.
9. Click **OK**.
10. You can continue to define more criteria and then save the criteria profile.
11. When you are done defining the criteria, save the criteria profile.
12. Select the criteria profile and then click **Activate** from the toolbar.✓

The status of the criteria profile is updated to show that it is now active. The criteria profile can now begin to intercept jobs according to the specifications of the criteria hierarchy and relaunch them as defined in the Tivoli Workload Scheduler job. If another criteria profile of the same criteria type was active, its status changes to inactive.

Using template files:
A template is a file with extension `.jdf` located in the same directory as the interception criteria file (*TWS_home*/`methods/r3batch_icp`). The template file contains instructions for the interception collector about how to run the intercepted SAP

R/3 job under control of Tivoli Workload Scheduler. Its syntax corresponds to the syntax of docommand in conman. You can use any text editor to maintain this file. Ensure that the user, LJUser, is able to read and write to this file.

If the user template file is empty, a template file named default.jdf is used. If default.jdf does not exist, the following instructions are used:

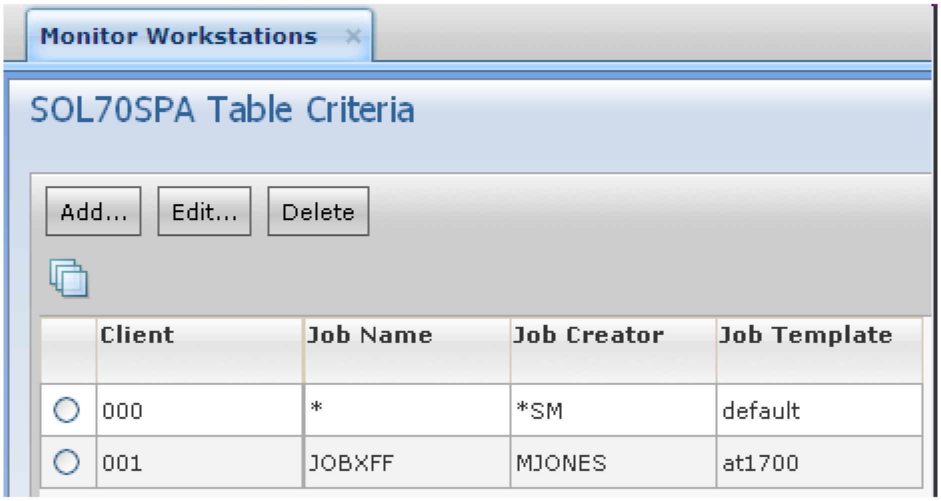
```
alias=SAP_$RUN_$JOBNAME_$JOBCOUNT
```

This means that the intercepted SAP R/3 jobs are to be restarted immediately, because of the absence of the at= job option. Their Tivoli Workload Scheduler names are composed of the string SAP_, the current run number of the interception collector, and the name and ID of the SAP R/3 job.

The instruction set for restarting an intercepted SAP R/3 job is retrieved in the following order:

1. From the template file, if an existing template is specified in the interception criteria file.
2. From the default template file, if the template is specified in the interception criteria file but does not exist, or if the template is not specified in the interception criteria file.
3. From the default instruction set, if the default template file does not exist.

Job interception example:
The following example demonstrates how different template files can be used to determine when an intercepted SAP R/3 job is restarted. The interception criteria table contains the following entries:



	Client	Job Name	Job Creator	Job Template
○	000	*	*SM	default
○	001	JOBXFF	MJONES	at1700

Figure 9. The Table Criteria panel

The table criteria specified, implies the following:

Client 000

All jobs started in client 000 by SAP R/3 users whose user name begins with sm, will be intercepted. The interception collector restarts the jobs using the instructions from the default template file default.jdf. If the default template file does not exist, then the SAP R/3 jobs are restarted immediately as specified in the default instruction set:

```
alias=SAP_$RUN_$JOBNAME_$JOBCOUNT
```

Client 001

The job named, JOBXFF, started in client 001 by SAP R/3 user named, MJONES, will be intercepted. The interception collector restarts the jobs using the instructions from the template file at1700.jdf. The SAP R/3 jobs are restarted at 17:00 with a random name, because of the alias command. The template file at1700.jdf contains the following entry:

```
alias;at=1700
```

Using placeholders: In the template files you can use a number of placeholders that are replaced by the interception collector at run time. They are listed in Table 42.

Table 42. Placeholders for job interception template files

Placeholder	Description
$CPU	Name of the extended agent workstation where the interception collector runs.
$CLIENT	Client number of the intercepted SAP R/3 job.
$JOBNAME	Name of the intercepted SAP R/3 job.
$JOBCOUNT	Job ID of the intercepted SAP R/3 job.
$USER	Name of the user who launched the SAP R/3 job.
$JOBNUM	Job number of the interception collector.
$RUN	Current run number of the interception collector.
$SCHED	Schedule name of the interception collector.
$RAND	Random number.

The template:

```
alias=ICP_$RAND_$JOBNAME_$JOBCOUNT_$CLIENT;at=1000
```

instructs the interception collector to restart the SAP R/3 job named DEMO_JOB with job ID 12345678 on client 100 at 10:00 as Tivoli Workload Scheduler job ICP_1432_DEMO_JOB_12345678_100.

Activating the job interception feature

Activate the job interception feature for the appropriate BC-XBP interface.

To enable the job interception feature: .

1. Run ABAP report INITXBP2. This report shows you the current status of the job interception and parent-child features, and allows you to toggle the status of both features.
2. Select the BC-XBP interface version as appropriate:
   - **Activate 3.0**
   - **Activate 2.0**
3. Save the changes.

The parent-child feature

In some situations, an SAP R/3 job dynamically spawns a number of other jobs; for example, to distribute the workload to the free application servers. Prominent examples are the mass activity jobs of the SAP R/3 FI-CA component. Before BC-XBP 2.0, it was difficult for external schedulers to handle this situation, because the business process does not usually end with the end of the initial job (parent job), but with the end of all subjobs (child jobs).

The BC-XBP 2.0 interface allows you to determine if a job has launched subjobs, together with their names and IDs, and so it is now possible to track them.

To activate this feature, use the `INITXBP2` ABAP report, which you can also use to toggle the status of job interception.

When the parent-child feature is active, Tivoli Workload Scheduler considers an SAP R/3 job as finished only after all its child jobs have ended. The status of the Tivoli Workload Scheduler job remains as `EXEC` while the parent job or any of its child jobs are running.

The status of the Tivoli Workload Scheduler job becomes `SUCC` if the parent job and all child jobs end successfully. If any of the jobs ended with an error, the status of the Tivoli Workload Scheduler job becomes `ABEND`.

Note: The parent-child feature can interfere with job interception because, although the parent job cannot be intercepted, any of its child jobs can be intercepted if they match the interception criteria. In this case, the Tivoli Workload Scheduler job remains in the `EXEC` status until the intercepted child job has been relaunched and has ended.

The joblogs of the child jobs are appended in the Tivoli Workload Scheduler `stdlist` after the joblog of the parent job.

Using Business Information Warehouse

Business Information Warehouse (BIW) is a data warehouse solution tailored to SAP R/3. It allows business reporting and decision support.

To use the InfoPackages component, you must have the SAP Business Warehouse Systems, version 2.0B or higher installed.

To use the Process Chains component, you must have the SAP Business Warehouse Systems, version 3.0B or higher installed.

The Support Package 9 (SAPKW31009) for SAP Business Warehouse version 3.1 is required so that `r3batch` can launch process chains.

Business Warehouse components

SAP R/3 supports two main Business Warehouse components, InfoPackages and Process Chains.

An InfoPackage is the entry point for the loading process from a specific **InfoSource** (a logical container of data source, generically named **InfoObject**). Technically, an InfoPackage is an SAP R/3 job whose aim is to load data. Like any other SAP R/3 job, it contains job-pecific parameters such as start time, and dependencies.

A Process Chain is a complex chain of different processes and their relationships. The processes within a process chain are not limited to data load processes, or InfoPackages, but also include:

- Attribute/Hierarchy Change run

- Aggregate rollup
- ABAP program
- Another process chain
- Customer build process

Defining user authorizations to manage SAP R/3 Business Warehouse InfoPackages and process chains

What you need to use SAP R/3 Business Warehouse InfoPackages and process chains.

Tivoli Workload Scheduler for Applications for SAP can manage SAP R/3 Business Warehouse InfoPackages and process chains. To use the SAP R/3 Business Warehouse functions, you must define a Tivoli Workload Scheduler user within SAP R/3 with full authorization for the ABAP Workbench object S_DEVELOP.

The user must also belong to the following profiles:

- S_BI-WHM_RFC (for Business Information Warehouse version 7.0, or later)
- S_RS_ALL
- Z_MAESTRO

Managing SAP R/3 Business Warehouse InfoPackages and process chains

You can manage existing InfoPackages and process chains on SAP systems from SAP.

Business Warehouse InfoPackages and process chains can only be created from the SAP R/3 environment. However, the Dynamic Workload Console supports pick lists of InfoPackages and process chains, so that you can also define Tivoli Workload Scheduler jobs for these existing objects.

You can create Tivoli Workload Scheduler job definitions that map to SAP jobs that already exist on SAP systems in the following environments:

- Distributed Distributed
- z/OS z/OS

The SAP jobs can run on extended agent workstations, dynamic agent workstations, dynamic pools, and z-centric workstations depending on the type of job definition you choose to create.

This section describes how to perform tasks such as creating the Tivoli Workload Scheduler job definitions that map to SAP jobs, how to display the details of these jobs, and how to rerun a process chain job.

Creating a Tivoli Workload Scheduler job that contains InfoPackages or process chains

Creating a job with InfoPackages or process chains.

This section describes how to create a Tivoli Workload Scheduler SAP job definition that references a Business Warehouse InfoPackage or Process Chain SAP job.

SAP job definitions can be created using both a distributed or z/OS engine and they can be scheduled to run on the following workstations with the r3batch access method:

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

Refer to the Dynamic Workload Console online help for a complete description of all UI elements for both engine types and all supported workstation types.

Take into consideration that:

- To be able to schedule InfoPackages using Tivoli Workload Scheduler, the scheduling options of the InfoPackage must have:
  - Start type set to **Start later in background process**.
  - Start time set to **Immediate**.
- To be able to control process chains using Tivoli Workload Scheduler, the scheduling options of the process chain must be **Start Using Meta Chain or API**. If the process chain is set to **Direct Scheduling**, it starts immediately when activated in the SAP system or transported to another SAP system.
- If you are using an operating system that does *not* support Unicode, set the **TWSXA_LANG** option. For details about the operating systems that support Unicode, see “Unicode support on r3batch” on page 79. For details about the **TWSXA_LANG** option, see “Setting National Language support options” on page 207.

You can create a SAP job definition to reference an InfoPackage or process chain using the Dynamic Workload Console.

The following procedure creates a Tivoli Workload Scheduler SAP job definition and references an InfoPackage or process chain in the Tivoli Workload Scheduler database:

1. Click **Tivoli Workload Scheduler** > **Workload** > **Design** > **Create Workload Definitions**.
2. Select a an engine, either distributed or z/OS. The **Workload Designer** is displayed.
3. From the Working List pane, click:
   - z/OS engine: **New** > **ERP**
   - Distributed engine: **New** > **Job Definition** > **ERP**
4. Select the SAP job definition in accordance with the engine and type of agent on which the job runs.

   z/OS engine

   SAP This job definition references an existing job on the SAP system and can run on dynamic agent workstations, dynamic pools, and z-centric workstations.

   Distributed engine

   SAP Job on Dynamic Workstations
   This job definition can run on dynamic agent workstations, dynamic pools, and z-centric workstations.

SAP Job on XA Workstations
This job definition can run on extended agent workstations. A workstation that is hosted by a fault-tolerant agent or master workstation.

5. In the Workspace pane, specify the properties for the job definition you are creating using the tabs available. The tabs for each type of SAP job definition are similar, but there are some differences depending on the type of engine you selected and the type of workstation on which the job runs. For more detailed information about the UI elements on each tab, see the Dynamic Workload Console online help.

   The **General** page requires information regarding the workstation that connects to the remote SAP system. If a default SAP connection is already configured, then these fields are alreaday prefilled, otherwise, you can specify the required information on the General page or you can configure a default connection to be used each time it is required in a definition, see "Setting the SAP data connection" on page 110 for more information.

   On the **Task** page, in **Subtype**, specify either **BW Process Chain** or **BW InfoPackage**.
6. Click **Save** to add the SAP job definition to the Tivoli Workload Scheduler database.

Task string to define Business Warehouse InfoPackages and process chain jobs

This section describes the task string parameters that control the running of the Business Warehouse InfoPackages and process chain jobs. You must specify them in the following places when you define their associated Tivoli Workload Scheduler jobs:

- If you use the Dynamic Workload Console, in the **R/3 command line** field of the Task page of the SAP job definition panel.
- As arguments of the `scriptname` keyword in the job definition statement, if you use the Tivoli Workload Scheduler command line.
- As arguments of the `JOBCMD` keyword in the `JOBREC` statement in the `SCRIPTLIB` of Tivoli Workload Scheduler for z/OS, if you are scheduling in an end-to-end environment.

The string syntax is the following:

Job definition syntax

```
►►── -job──job_name── -i──┬─ipak_───┬──┬─────────┬──┬─────────┬──────────────►
                          └─pchain_─┘  └─ -debug─┘  └─ -trace─┘

►──┬──────────────────┬──┬─────────────────────────────────┬──────────────────►
   └─ -flag──┬─imm───┬┘  └─ -flag──┬─enable_pchainlog──┬───┘
             └─immed─┘             └─disable_pchainlog─┘

►──┬───────────────────────────────┬──┬──────────────────────────────────┬────►
   └─ -flag──┬─enable_ipaklog──┬───┘  └─ -flag──┬─level_all_pchainlog─┬─┘
             └─disable_ipaklog─┘                └─level_n_pchainlog───┘
```

```
►─┬──────────────────────────────────────────────┬──────────────────────────►
  └─ -flag──┬─pchainlog_chains_only─────────────┤
            ├─pchainlog_chains_and_failed_proc──┤
            └─pchainlog_complete────────────────┘

►─┬────────────────────────────────────┬────────────────────────────────────►
  └─ -flag──┬─enable_pchainlog_bapi_msg──┤
            └─disable_pchainlog_bapi_msg─┘

►─┬───────────────────────────────────┬─┬──────────────────────────────┬───►◄
  └─ -flag──┬─enable_pchain_details───┤ └─ -flag──┬─pchain_rerun─────┤
            └─disable_pchain_details──┘           ├─pchain_restart───┤
                                                  └─pchain_refresh───┘
```

The parameters are described in Table 37 on page 118.

Table 43. Task string parameters for SAP R/3 jobs

Parameter	Description	GUI Support
-job *job_name*	The name of the task to be run. It is either an InfoPackage technical field name, or a process chain name. This parameter is mandatory.	✓
-i {ipak_ | pchain_}	One of the following: **ipak_** Target job is an InfoPackage **pchain_** Target job is a process chain	✓
-debug	Turns on the most verbose `r3batch` trace. This option is for debugging the extended agent and should not be used in standard production.	✓
-trace	Turns on the `SAP RFC` trace. When you use this option, a trace file is created in the Tivoli Workload Scheduler `methods` directory. In UNIX, this trace file is called `dev_rfc`. In Windows, the file is called `rfc`*xxxxx_xxxxx*`.trc`. This option is for debugging the extended agent and should not be used in standard production. Ensure that you delete the trace option from the job after you have performed debug procedures. The trace file can become very large and unmanageable.	✓
-flag {imm | immed}	Specifies to launch the job immediately, meaning that if there are no spare work processes, the job fails.	✓
-flag {enable_pchainlog | disable_pchainlog}	Enables or disables retrieval and appending of the process chain job log in the Tivoli Workload Scheduler `stdlist`. Disable if the size of the log affects performance. A related configuration option can be set for this purpose at a more general level. See Table 35 on page 95.	✓
-flag {enable_ipaklog | disable_ipaklog}	Enables or disables retrieval and appending of the InfoPackage job log in the Tivoli Workload Scheduler `stdlist`. Disable if the size of the log affects performance. A related configuration option can be set for this purpose at a more general level. See Table 35 on page 95.	

Table 43. Task string parameters for SAP R/3 jobs (continued)

Parameter	Description	GUI Support
-flag {level_*n*_pchainlog | level_all_pchainlog}	Allows for retrieval of process chain logs down to the process chain level you specify. **level_*n*_pchainlog** Specifies that the process chains are logged down to, and including, the level represented by number *n*. **level_all_pchainlog** Specifies that all the process chains are logged. The default is `level_1_pchainlog`. A related configuration option can be set for this purpose at a more general level. See Table 35 on page 95.	
-flag {pchainlog_chains_only | pchainlog_chains_and_failed_proc | pchainlog_complete}	Specifies what type of process chain-related logs will be retrieved. **pchainlog_chains_only** Only the process chains are logged. **pchainlog_chains_and_failed_proc** In addition to the process chains, all the processes that failed are also logged. **pchainlog_complete** The process chains and all processes are logged. The default is **pchainlog_complete**. A related configuration option can be set for this purpose at a more general level. See Table 35 on page 95.	
-flag {enable_pchainlog_bapi_msg | disable_pchainlog_bapi_msg}	Enables or disables retrieval of additional messages from the BAPI calls from the SAP Business Warehouse process chains and appends them to the Tivoli Workload Scheduler `stdlist`.	
-flag {enable_pchain_details | disable_pchain_details}	Enables or disables the display of details about the process chain job. A related configuration option can be set for this purpose at a more general level. See Table 35 on page 95.	✓

Table 43. Task string parameters for SAP R/3 jobs (continued)

Parameter	Description	GUI Support
-flag {pchain_rerun | pchain_restart | pchain_refresh}	Determines the action that Tivoli Workload Scheduler performs when you rerun a job that submits a process chain. **pchain_rerun** Tivoli Workload Scheduler creates another process chain instance and submits it to be run again. **pchain_restart** Tivoli Workload Scheduler restarts the original process chain from the failing processes to the end. **pchain_refresh** Tivoli Workload Scheduler updates the status and details of the original process chain. For more details about rerunning a process chain, refer to "Rerunning a process chain job" on page 167.	✓

Note: Typically, the `-debug` and `-trace` options are for debugging the extended agent and should not be used in standard production.

The following is an example for an InfoPackage job whose technical field name is `ZPAK_3LZ3JRF29AJDQM65ZJBJF5OMY`:

```
-job ZPAK_3LZ3JRF29AJDQM65ZJBJF5OMY -i ipak_
```

Displaying details about Business Warehouse InfoPackages

To display details about a Business Warehouse InfoPackage, perform the following steps:

1. Open the **Workload Designer**, from the portfolio, click **Tivoli Workload Scheduler** > **Workload** > **Design** > **List Jobs on SAP**.
2. In **Engine name**, select the name of the Tivoli Workload Scheduler engine connection from which you want to view SAP job details.
3. In **SAP Job Type**, select **Business Warehouse InfoPackage**.
4. In **Workstation name**, specify the workstation where the SAP job runs. If you do not know the object name, click the **...** (Browse) button. In the **Name** and **Location** panel, enter some characters of the object name (asterisk is supported as a wildcard) and click **Start**. From the displayed list, select the workstation you want to use, and click **OK**.
5. Click **Display**. The list of available jobs of type Business Warehouse InfoPackage for the specified engine is displayed.
6. Select the job for which you want to display the details and click **Details**.
7. When you have finished viewing the details for the job, click **OK** to return to the list of SAP jobs on the workstation specified.

Displaying details about a process chain job

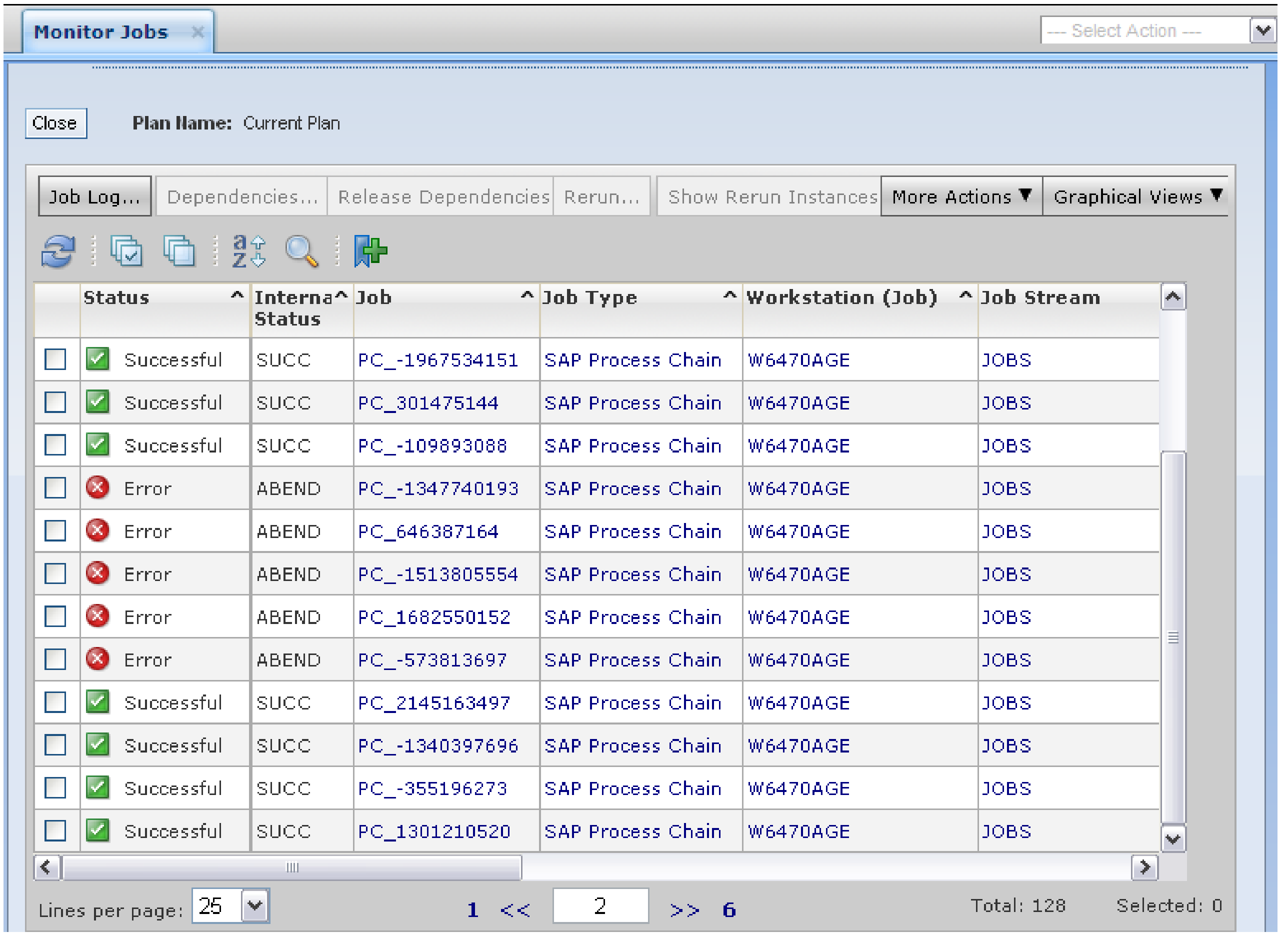
You can view the details for a process chain job including any local subchains contained in the process chain.

Ensure you have performed the following steps before running this procedure:

- Set the **pchain_details** option to `ON` in the common options file. For more information about this option, refer to "Defining the common options" on page 94.
- Distributed In a distributed environment, customize the Browse Jobs tasks that you created *before* installing Tivoli Workload Scheduler for Applications 8.4 Fix Pack 1 to show the **Job Type** column. For details about how to customize the task properties, refer to the Dynamic Workload Console online help.
- In a z/OS environment, you must customize the task properties to display the **Advanced Job Type** column that indicates the job type. For details about how to customize the task properties, refer to the Dynamic Workload Console online help.

To display details about an SAP Process Chain that you scheduled as a Tivoli Workload Scheduler job, perform the following steps from the Dynamic Workload Console.

1. Click **Workload** > **Monitor Jobs**.
2. The list of defined tasks or queries for monitoring jobs is displayed. Select the hyperlink of a task to display the related list of jobs. If you have a predefined task to display SAP jobs or process chain jobs, click that task.
3. In **Engine name**, select the name of the Tivoli Workload Scheduler engine connection from where you want to work with SAP jobs and click **OK**.
4. The table of results for the task is displayed:



Plan Name: Current Plan

	Status	Internal Status	Job	Job Type	Workstation (Job)	Job Stream
□	✓ Successful	SUCC	PC_-1967534151	SAP Process Chain	W6470AGE	JOBS
□	✓ Successful	SUCC	PC_301475144	SAP Process Chain	W6470AGE	JOBS
□	✓ Successful	SUCC	PC_-109893088	SAP Process Chain	W6470AGE	JOBS
□	☒ Error	ABEND	PC_-1347740193	SAP Process Chain	W6470AGE	JOBS
□	☒ Error	ABEND	PC_646387164	SAP Process Chain	W6470AGE	JOBS
□	☒ Error	ABEND	PC_-1513805554	SAP Process Chain	W6470AGE	JOBS
□	☒ Error	ABEND	PC_1682550152	SAP Process Chain	W6470AGE	JOBS
□	☒ Error	ABEND	PC_-573813697	SAP Process Chain	W6470AGE	JOBS
□	✓ Successful	SUCC	PC_2145163497	SAP Process Chain	W6470AGE	JOBS
□	✓ Successful	SUCC	PC_-1340397696	SAP Process Chain	W6470AGE	JOBS
□	✓ Successful	SUCC	PC_-355196273	SAP Process Chain	W6470AGE	JOBS
□	✓ Successful	SUCC	PC_1301210520	SAP Process Chain	W6470AGE	JOBS

Lines per page: 25

Total: 128 Selected: 0

Figure 10. Dynamic Workload Console - Table of results

5. Select a process chain job. For each process chain job, a hyperlink named **SAP Process Chain** is displayed.

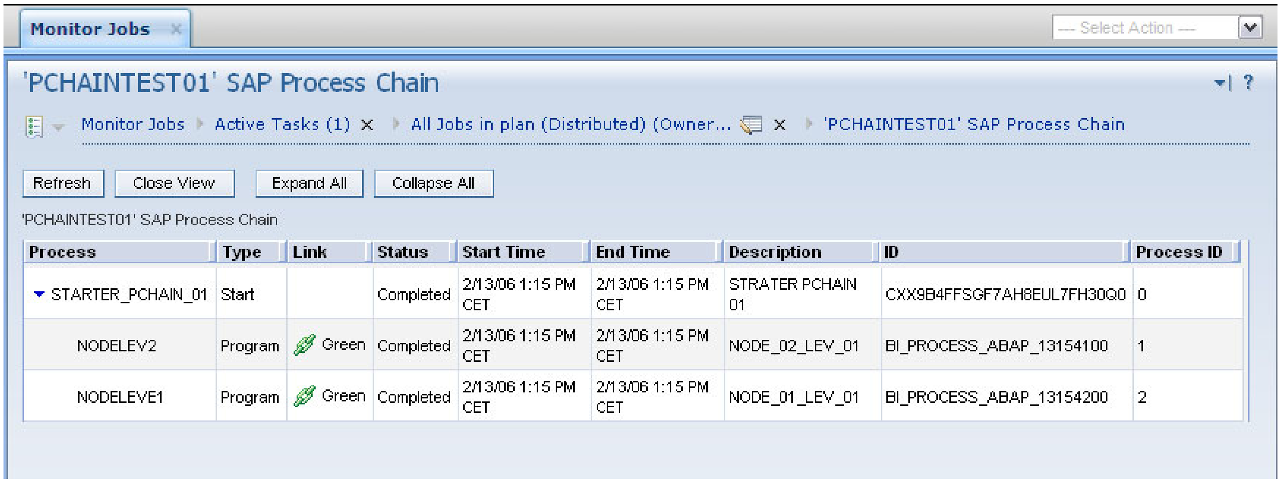
   Distributed **Distributed environment**

   The **Job Type** column displays **SAP Process Chain** to help you identify SAP process chain jobs.

   z/OS **z/OS environment**

   The **Advanced Job Type** column displays **SAP Process Chain** to help you identify SAP process chain jobs.

   Click the hyperlink for the job whose details you want to display.
6. The details for the process chain are displayed:



'PCHAINTEST01' SAP Process Chain

'PCHAINTEST01' SAP Process Chain

Process	Type	Link	Status	Start Time	End Time	Description	ID	Process ID
STARTER_PCHAIN_01	Start		Completed	2/13/06 1:15 PM CET	2/13/06 1:15 PM CET	STRATER PCHAIN 01	CXX9B4FFSGF7AH8EUL7FH30Q0	0
NODELEV2	Program	Green	Completed	2/13/06 1:15 PM CET	2/13/06 1:15 PM CET	NODE_02_LEV_01	BI_PROCESS_ABAP_13154100	1
NODELEVE1	Program	Green	Completed	2/13/06 1:15 PM CET	2/13/06 1:15 PM CET	NODE_01_LEV_01	BI_PROCESS_ABAP_13154200	2

Figure 11. Dynamic Workload Console - Details of a process chain job

Tivoli Workload Scheduler for Applications monitors the process chain job until the job completes. The details shown reflect the last monitoring process performed. Perform a restart of the process chain indicating a refresh operation to synchronize the details with those on the remote SAP system to have the most updated information possible..

If the process chain contains local subchains, a hyperlink is displayed for each one. Click the hyperlink you want, to display details about the corresponding subchain job.Alternatively, you can display the process chain details by clicking the hyperlink for the job and display the job properties panel. Click the hyperlink shown under **SAP Job Details**. The details for the process chain are displayed.

Rerunning a process chain job

Process chain jobs can be rerun from the start, rerunning the entire process chain, or they can be restarted from a specific process. Restarting a process enables you to restart without rerunning the whole process chain again. You can choose to either restart from the failed processes in a process chain, or restart a specific process indicating the related process ID.

To rerun an SAP job that submits a process chain, you can use one of the following user interfaces:

conman
For details, refer to the *Tivoli Workload Scheduler User's Guide and Reference*.

Dynamic Workload Console
See "Procedure for rerunning a process chain job" on page 171 for information about performing this task from the console.

For information about rerunning an SAP Standard R/3 job, see "Rerunning a standard SAP job" on page 129.

In general, when you rerun a process chain job, the new job is assigned the name of the alias you specify. To keep the original job name, set the Tivoli Workload Scheduler global option **enRetainNameOnRerunFrom** to yes. For details about this option, see *Tivoli Workload Scheduler Administration Guide*.

On extended agents, an alias is mandatory for each action you perform on the process chain job and the action itself, is the prefix of the alias name. For example, if you choose to restart a process chain from the failed processes, and assign PCHAIN1 as the alias for the process chain job, then the new job name is `Restart_PCHAIN1`.

z/OS In a z/OS environment, the process chain job maintains the same name and the Monitor Jobs view always displays the status for the last action performed on the job. Every time a rerun is performed on a process chain job, a new instance is generated each with a different ID.

Note:

1. By default, if you do not specify any setting, rerunning a process chain job corresponds to submitting a new process chain instance.
2. If you kill a Tivoli Workload Scheduler job that submits a process chain, the process chain is removed from schedule in the SAP Business Information Warehouse system. To restart the same process chain instance with r3batch, you require at least the following SAP Business Information Warehouse versions:
   - 3.0 with SP25
   - 3.1 with SP19
   - 3.5 with SP10
   - 7.0

   If your version of SAP Business Information Warehouse is earlier, you can restart the process chain only manually, through the SAP graphical interface.

   Table 44 shows the action performed when you rerun a Tivoli Workload Scheduler job that submits a process chain, depending on the settings you specify. These are the actions performed when you submit the rerun operation using the Rerun button from the Monitor Jobs view.

Table 44. Actions performed when you rerun a process chain job

Action performed	Description and setting
A new process chain instance is submitted	Tivoli Workload Scheduler creates another process chain instance and submits it to be run again. This action occurs when: • On extended agents, you specify `RERUN`*`value`* as the step to rerun, where *`value`* is any value you want. This setting overrides the settings in the job definition and options file, if any. In an end-to-end environment, you can perform this action on a centralized job by adding the following parameter to the script file: `-flag pchain_rerun` • In the job definition, you set `-flag pchain_rerun`. This setting overrides the setting in the options file, if any. For a description of this parameter, see Table 43 on page 162. • In the options file, you set the **pchain_recover** option to `rerun`. For a description of this option, refer to Table 35 on page 95.

Table 44. Actions performed when you rerun a process chain job (continued)

Action performed	Description and setting
The original process chain is rerun from the failed processes	Tivoli Workload Scheduler restarts the original process chain from the failed processes to the end. In this way, after you detected the error that caused the failure and performed the recovery action, you can rerun the process chain job from the failed processes and have its run completed. This action is performed only if at least one process in the process chain did not complete successfully. It occurs when: • On extended agents, you specify `RESTART`*`value`* as the step to rerun, where *`value`* is any value you want. This setting overrides the settings in the job definition and options file, if any. In an end-to-end environment, you can perform this action on a centralized job by adding the following parameter to the script file: `-flag pchain_restart` • In the job definition, you set `-flag pchain_restart`. This setting overrides the setting in the options file, if any. For a description of this parameter, see Table 43 on page 162. • In the options file, you set the **pchain_recover** option to `restart`. For a description of this option, refer to Table 35 on page 95.

Table 44. Actions performed when you rerun a process chain job (continued)

Action performed	Description and setting

The process that you specify is restarted	Tivoli Workload Scheduler restarts the process of the original process chain that you specify, and monitors the process chain run until its final state.
	On extended agents, this action occurs when you specify `PROCESS`*`processID`* as the step to rerun, where *`processID`* is the identifier of the process you want. For example, if the process ID is 3, you must specify `PROCESS3` as the step.
	You can view the process IDs in the following ways:
	• Dynamic Workload Console, version 8.5 or later. From the panel where the details about the process chain are displayed, see the column named Process ID. For details about how to display the process chain details, refer to "Displaying details about a process chain job" on page 164.
	• Tivoli Workload Scheduler job log, as follows:
	`+++ EEWO1071I Start of process chain PCHAIN1`
	`Process Chain PCHAIN1 (Log ID:D3C0ZWAYESD58PXOYPEOGNZK7).`
	`-------------------------`
	`Process Type: TRIGGER.`
	`Process Variant: PCHAIN1_STARTER.`
	`Actual State: F.`
	`...`
	`>>` **`Process ID: 3.`**
	`   Process Type: ABAP.`
	`   Process Variant: Z_PCHAIN1_NODE3.`
	`   Actual State: F.`
	`   Instance: D3C0ZXL3IJ8LR5O9Q1D9A4Y4N.`
	`>>` **`Process ID: 4.`**
	`   Process Type: ABAP.`
	`   Process Variant: Z_PCHAIN1_NODE1.`
	`   Actual State: .`
	`   Instance: D3C0ZZKS0RR88DKRJQ09Z1WW7.`
	`+++ EEWO1072I End of process chain PCHAIN1`
	The following list shows the meaning of the alphabetic value used as the actual state in the job log:
	Actual state — **Meaning**
	A Active
	F Completed
	G Successfully completed
	P Planned
	Q Released
	R Ended with errors
	S Skipped
	X Canceled
	Y Ready
	blank Undefined
	In an end-to-end environment, you can perform this action on a centralized job by adding the following parameter to the script file:
	`-pchain_pid` *`processID`*

Table 44. Actions performed when you rerun a process chain job (continued)

Action performed	Description and setting
The status and details of the original process chain are updated	Tivoli Workload Scheduler monitors the original process chain until its final status. This action occurs when: • On extended agents, you specify `REFRESH`*`value`* as the step to rerun, where *`value`* is any value you want. This setting overrides the setting in the job definition, if any. In an end-to-end environment, you can perform this action on a centralized job by adding the following parameter to the script file: `-flag pchain_refresh` • In the job definition, you set `-flag pchain_refresh`. For a description of this parameter, see Table 43 on page 162.

Procedure for rerunning a process chain job:

You can rerun all of the processes in the process chain from the Dynamic Workload Console or you can rerun at a process level.

z/OS In z/OS environments, you need to set the status of the job to **Ready** before you can rerun the job.

1. Select a job and click **Set Status**.
2. In **Change Status**, select **Ready**.
3. Click **OK** to return to the list of jobs.

To rerun a process chain SAP job, perform the following steps:

1. Click **Workload** > **Monitor Jobs**.
2. The list of defined tasks or queries for monitoring jobs is displayed. Select the hyperlink of a task to display the related list of jobs. If you have a predefined task to display SAP jobs or process chain jobs, click that task.
3. In **Engine name**, select the name of the Tivoli Workload Scheduler engine connection from where you want to work with SAP jobs and click **OK**.
4. A list of jobs is displayed. Select a process chain job.

   Distributed **Distributed environment**

   The **Job Type** column displays **SAP Process Chain** to help you identify SAP process chain jobs.

   z/OS **z/OS environment**

   The **Advanced Job Type** column displays **SAP Process Chain** to help you identify SAP process chain jobs. To display the **Advanced Job Type** column in the table, edit the **Task Properties** and in **Column Definition**, add the **Advanced Job Type** column to the **Selected Columns** list. Move the column up to define the order of the column in the table and make it more visible.
5. Rerun the job.
   a. Click **More Actions** > **Restart Process Chain**.
   b. Select the action you want to perform on the selected process chain:

      Rerun Reruns the entire process chain. The process chain ID on the SAP system remains the same, as well as the job identifier on z/OS systems.

Distributed Specify an alias to identify the new job. In distributed systems the rerun process chain is identified with this alias name prefixed by `RERUN`.

Refresh

Refreshes the Dynamic Workload Console view with the latest updates on the remote SAP system so that the two views are synchronized.

Distributed Specify an alias to identify the new job. In distributed systems the refreshed process chain is identified with this alias name prefixed by `REFRESH`.

Restart from the failed processes

Action available only for process chains in error state. Rerun only some steps of the process chain, starting from the failed processes.

Distributed Specify an alias to identify the new job. In distributed systems the restarted process chain is identified with this alias name prefixed by `RESTART`.

Restart from a specific process

Action available only for process chains in error state. Rerun only some steps of the process chain, starting from the process specified in the **SAP Process ID** field. You can find the process ID by opening the job log or viewing the job type details from the table of results of your monitor job task.

Distributed In distributed systems the restarted process chain is identified with this alias prefixed by `PROCESS`.

6. Click **OK** to perform the selected action on the process chain.

The job reruns immediately.

Business scenario: rerunning the original process chain job from the failed process: As a scheduling administrator, you are responsible for managing batch jobs in both SAP and non-SAP systems. The workflow is one or more job streams in Tivoli Workload Scheduler. A job stream contains jobs that collect and prepare data for month-end closing over all sales channels. The month-end closing report requires data to be collected from several sales and distribution systems. Data is collected using local and remote process chains in the SAP Business Intelligence system. The process chains include a set of Infopackages, ABAP reports, and operating system jobs to sort the report data by a logical hierarchy.

To administer from a single point of control, you link the SAP process chains to Tivoli Workload Scheduler through Tivoli Workload Scheduler for Applications.

During batch processing, a Tivoli Workload Scheduler job comprising a process chain, failed. Optionally, you can see which processes failed either from the Dynamic Workload Console (for details, see “Displaying details about a process chain job” on page 164) or in the job log. You ask the SAP administrator to fix the cause of the error, then, on an extended agent, you rerun the Tivoli Workload Scheduler job by setting the step as `RESTART`*`value`*. In this way, the original process chain is restarted from the failed processes and continues until the ending step.

Alternatively, you can select the process chain job from the Monitor Jobs view on the Dynamic Workload Console and then select **More Actions** > **Restart Process Chain** and then select the **Restart from the failed processes** option.

Business scenario: restarting a specific process of the process chain: You might decide to restart a single process as a preparation step before restarting the failed processes of a process chain. A failed process might have corrupted some data, so you run the single process to restore the data and set up the required system state before you rerun the other processes in the process chain.

Suppose you are using InfoPackages and process chains to extract data from one or several sources and you want to transform this data into managerial reports, for example by using aggregate functions. If the process that transforms this data fails, it might corrupt the data that the preceding InfoPacakge process had successfully extracted. After fixing the problem with the transformation process, you must restart the InfoPackage extraction process to reload the data, even though this extraction process had completed successfully before. Restart the failed transformation process only after the data has been reloaded, either by restarting the failed processes of the process chain or by restarting just the failed transformation process.

On an extended agent, from the Monitor Jobs view on the Dynamic Workload Console, select the process chain and click Rerun, then specify `PROCESS`*`processID`* as the step to rerun, where *processID* is the identifier of the process you want to restart.

To restart a specific process of the process chain, from the Monitor Jobs view on the Dynamic Workload Console, select the process chain and click **More Actions** > **Restart Process Chain** and then select the **Restart from a specific process** option, specifying the process ID in the **SAP Process ID** field.

Job throttling feature

This chapter describes how the job throttling feature helps you to improve the efficiency of your scheduling on SAP systems and reduce the batch window for your SAP jobs to a minimum. Using advanced XBP 2.0 and 3.0 functions, such as the job interception and parent-child, the job throttler ensures that the SAP system is not overloaded and the number of released jobs does not exceed the total number of SAP background work processes in the system.

You can also configure the job throttler to send data related to its activity to the SAP Computing Center Monitoring System (CCMS) for monitoring purposes.

Business scenario

You manage your Internet sales through an application software that verifies that data is correct, checks the availability of the item, and validates the order. To process all the orders received, you scheduled a Tivoli Workload Scheduler job to run every 12 hours, connect to SAP, and generate a child job for every order to process. Child jobs are in charge of creating shipping bills, checking destination address, and forwarding the orders to the appropriate carrier, thus optimizing the delivery process. A potential overload of the system might occur during peak times, for example over Christmas, and could risk the late delivery of orders, damaging your business. To manage the submission of jobs and activate an advanced management of their priority class (for both parent and child jobs), enable the job throttling feature.

Additionally, you might want to set a policy so that an SAP CCMS alert is raised each time the number of jobs to be released under the control of the job throttler exceeds a certain threshold. To do this, you enable the job throttler to send data to the SAP CCMS monitoring architecture. At job throttler startup, an MTE that monitors the number of jobs to be released by the job throttler is created. By including the MTE in a monitoring set and specifying the related threshold, you are alerted each time the threshold is exceeded.

Software prerequisites

To use job throttling, you must have the SAP JCo 3.0.2 libraries (`dll` and `jar` files) installed in the *`TWS_home`*`/methods/throttling/lib` directory. To download JCo 3.0.2, visit the following Web site: http://service.sap.com/connectors/

JCo libraries are supported on the following operating systems (for the complete list of the operating systems supported by JCo, refer to SAP Note 1077727):

- AIX 5.2, 5.3, 6.1 (64-bit)
- HP-UX PA-RISC, versions HP-UX B11.11, B11.23, B11.31 (64-bit)
- HP-UX Itanium, versions HP-UX B11.23, B11.31 (64-bit)
- Linux x86 32-bit, versions SuSE SLES9, SuSE SLES10, RedHat EL4, RedHat EL5
- Linux x86 64-bit, versions SuSE SLES9, SuSE SLES10, RedHat EL4, RedHat EL5
- Linux Itanium 64-bit, versions SuSE SLES9, SuSE SLES10, RedHat EL4, RedHat EL5
- Linux zSeries 64-bit, versions SuSE SLES9, SuSE SLES10, RedHat EL5
- Linux PowerPC® 64-bit, versions SuSE SLES9, SuSE SLES10, RedHat EL4, RedHat EL5
- Solaris SPARC 64-bit, versions 9, 10
- Solaris x64, version Solaris 10
- Windows x86 32-bit, versions Windows XP, Windows Server 2003, Windows Vista
- Windows x64 64-bit, versions Windows Server 2003, Windows Vista, Windows Server 2008
- Windows Itanium 64-bit, versions Windows Server 2003, Windows Server 2008

Setting and using job throttling

The job throttler enqueues intercepted jobs and releases them when the background work processes that they need on the SAP server or SAP server group are available. The queue of intercepted jobs is sorted by scheduling time and priority of SAP jobs. When the SAP parent-child feature is enabled, child jobs inherit their progenitor's priority so that new urgent jobs are run before other planned jobs.

The following sections describe the steps to operate job throttling.

Step 1. Setting the options in the options file

To define the behavior of the job throttling feature, set the following options in the options file. For detailed information about the options, see Table 35 on page 95.

- `throttling_enable_job_class_inheritance`
- `throttling_enable_job_interception`
- `throttling_interval`
- `throttling_max_connections`
- `throttling_release_all_on_exit`

Step 2. Enabling and configuring the job interception feature

As a prerequisite, the job throttler requires that the job interception feature is enabled in the SAP system. To enable and configure job interception, follow these steps.

Note: Ensure that the job throttling and job interception features are not running at the same time. The job throttler cannot start if interception collector jobs are running.

1. Enable job interception, either automatically or manually, as follows:

   Automatic activation (meaning that the job throttler enables the job interception on SAP system)

   In the options file of the workstation with the r3batch access method you are using, set `throttling_enable_job_interception=on` (this is the default).

   Manual activation

   a. In the SAP system, run the INITXBP2 ABAP program in the transaction `se38` and enable job interception.
   b. In the options file of the workstation with the r3batch access method you are using, set `throttling_enable_job_interception=off`.

   Note: When you stop the job throttler, the setting for the job interception feature that was previously configured on the SAP system is restored.
2. In the SAP system, configure the job interception criteria as follows:
   a. Launch the transaction `se16` to access the table TBCICPT1, where the interception settings are maintained.
   b. Set the job name, creator, and client related to the jobs you want to intercept. To intercept all SAP jobs, specify the wildcard * (asterisk) for the job name, creator, and client.
   c. Save your settings and close the dialog.

SAP will intercept all the jobs matching the selection criteria, and the job throttling will release all the jobs that were intercepted.

Step 3. Enabling job class inheritance

You can configure the job throttler to have the intercepted job inherit the priority class from its progenitor (the top-level job in the hierarchy), if the progenitor class is higher than the intercepted job class. To do this, in the options file set `throttling_enable_job_class_inheritance=on`; this setting automatically enables the parent-child feature on the SAP system.

Note: When you stop the job throttler, the setting for the parent-child feature that was previously configured on the SAP system is restored.

Step 4. Configuring the logging properties

You can configure the trace properties of the job throttler by editing the logging configuration file `jobthrottling.properties` located in *TWS_home*`/methods/throttling/properties`.

To configure the trace level, follow the procedure.

1. Set the trace level property. The supported trace levels are: `DEBUG_MIN`, `DEBUG_MID`, and`DEBUG_MAX`, where `DEBUG_MAX` is the most verbose trace level.
2. Save the changes.

When making changes to the trace level setting, the changes are effective immediately after saving the .properties file. Other changes might require a restart to make them effective.

You can also configure the name, number, and size of the trace file. By default, the job throttler generates a maximum of 3 files of 5 MB in the `TWS_home/methods/traces` directory.

Note: The job throttler creates the `TWS_home/methods/traces` directory as soon as it is started.

Step 5. Starting and stopping the job throttling feature

To start job throttling, run the `jobthrottling` executable file related to the operating system you are using. Optionally, you can create a Tivoli Workload Scheduler job that starts the job throttler.

Note: On Windows systems using a single-byte character language, to start job throttling from a command prompt ensure that the DOS shell font is *not* Lucida Console. Ensure also that you set the Tivoli Workload Scheduler environment by entering the following command:

```
TWS_home\tws_env.cmd
```

From a command prompt, enter:

UNIX operating systems

```
TWS_home/methods/jobthrottling.sh {XAname|base_options_filename}
[-scratch]
```

Windows operating systems

```
TWS_home\methods\jobthrottling.bat {XAname|base_options_filename}
[-scratch]
```

Where:

XAname
: The name of the extended agent you are using.

base_options_filename
: For dynamic and z-centric agents, the file name of the options file without the extension, defined on the engine workstation hosting the workstation with the r3batch access method.

-scratch
: If you enabled the job throttler to send data to CCMS (for details, see “Sending data from job throttling to the CCMS Monitoring Architecture” on page 177), the job throttler starts and resets the attribute MTE named `JT total released jobs` to 0. If you do not specify **-scratch**, the job throttler starts and increments the `JT total released jobs`.

  This parameter is optional, and has effect only if the job throttler sent its data to CCMS at least once before.

To know the syntax for the `jobthrottling` command, run the command as follows:

```
►►—jobthrottling— -u————————————————————————————————————————►◄
```

To stop the job throttler, enter the following command (optionally, you can create a Tivoli Workload Scheduler job that stops the job throttler):

UNIX operating systems

```
TWS_home/methods/stop-jobthrottling.sh
{XAname|base_options_filename}
```

Windows operating systems

```
TWS_home\methods\stop-jobthrottling.bat
{XAname|base_options_filename}
```

Alternatively, you can enter the following command (you must be connected as *TWSUser* and have read and write permissions on the `txt` file):

```
echo shutdown > TWS_home/methods/{XAname|base_options_filename}_jobthrottling_cmd.txt
```

The job throttler stops:

- When the timestamp of {*XAname*|*base_options_filename*}`_jobthrottling_cmd.txt` is later than the time when the job throttler started.
- Within the time interval you specified in the `throttling_interval` option.

Sending data from job throttling to the CCMS Monitoring Architecture

You can configure the job throttler to send data related to its activity to the SAP Computing Center Monitoring System (CCMS) for monitoring purposes. Sending data from the job throttler to CCMS is supported if you have at least the SAP Web Application Server 6.20, Support Package 12 installed.

In the options file, set the following options (for details, see Table 35 on page 95):

```
throttling_send_ccms_data
throttling_send_ccms_rate
```

In this way, at job throttler startup the following monitoring tree elements (MTE) are created:

- A context MTE named **ITWS for Apps**.
- An object MTE with the same name as the Tivoli Workload Scheduler for Applications extended agent where the job throttler is running. This object MTE belongs to the context MTE **ITWS for Apps**.
- The following attribute MTEs:

  JT total released jobs

  The total number of jobs that the job throttler has released since startup. This value depends on the `-scratch` option you set at job throttler startup; for details, see "Step 5. Starting and stopping the job throttling feature" on page 176.

  JT queue

  The number of enqueued intercepted jobs to be released.

  JT released jobs per cycle

  The number of released jobs in the latest run. This value depends on the `throttling_send_ccms_rate` setting; for details, see Table 35 on page 95.

Note: By default `throttling_release_all_on_exit` is set to `ON`, meaning that when you stop the job throttler, all the intercepted jobs are released. However, these jobs are not considered when updating the **JT total released jobs**, **JT queue**, and **JT released jobs per cycle** MTEs.

To begin monitoring, include the MTEs in the monitoring set you want, and set the thresholds to generate an alert.

You can define a Tivoli Workload Scheduler event rule based on the CCMS alerts; for detailed information, refer to "Defining event rules based on CCMS Monitoring Architecture alerts" on page 197.

For example, to define an event that monitors the attribute MTE **JT total released jobs**, on the extended agent workstation named `SAP_XA`, connected to the SAP system ID `T01`, specify the following information:

XA Workstation
: `SAP_XA`

MTE SAP System ID
: `T01`

MTE Monitoring Context Name
: `ITWS for Apps`

MTE Monitoring Object Name
: `SAP_XA`

MTE Monitoring Attribute Name:
: `JT total released jobs`

Deleting the monitoring tree elements

After you stopped the job throttling feature, if you configured it to send its status data to CCMS, you can delete one or more MTEs that were created. To do this:

1. From the SAP GUI, invoke the transaction `rz20` to display a list of monitor sets.
2. Locate the monitor set named SAP CCMS Technical Expert Monitors, and expand it.
3. Locate the monitor named All Monitor Contexts, and double-click it to open it.
4. From the action menu, select **Extras -> Activate Maintenance Functions**.
5. Locate the MTE named **ITWS for Apps** and select it.
6. Right-click the MTE and select **Delete**. You are prompted to choose one of the delete options.
7. Select the option you want. The MTE is deleted accordingly.

Note: Deleting **ITWS for Apps** from the All Monitor Contexts monitor, deletes also all the copies that you might have created in other monitors.

Exporting SAP R/3 factory calendars

This chapter describes how to export SAP R/3 factory calendars into a file whose format can be processed by the Tivoli Workload Scheduler **composer** command line, to add the exported calendar definitions to the Tivoli Workload Scheduler database.

Business scenario

You might want to configure your Tivoli Workload Scheduler scheduling activities based on the schedule calendar in your SAP R/3 system. To do this, use the `r3batch` export function to export the SAP R/3 calendar definitions into a file whose format is compatible with the Tivoli Workload Scheduler **composer** command line. Based on the parameters you specify, you create a file that contains only the SAP R/3 calendar definitions that meet your scheduling requirements. Use this file as input for the **composer add** command, to import the calendar definitions into the Tivoli Workload Scheduler database. Your Tivoli Workload Scheduler and SAP R/3 calendars are now synchronized.

To keep the Tivoli Workload Scheduler and SAP R/3 calendar definitions synchronized and avoid duplicating data maintenance in the two environments, you can schedule to export the calendar definitions from SAP R/3 and import them to Tivoli Workload Scheduler on a regular basis using a dedicated job.

Exporting and importing SAP R/3 factory calendars

Refer to the following sections:

- “Exporting factory calendars” for an explanation about how you use the **r3batch** export function to access and download factory calendars available in an SAP R/3 system. The main purpose of this function is to create an output file that can be used by the **composer** to synchronize Tivoli Workload Scheduler calendars with existing SAP R/3 factory calendars, integrating the calendar definitions from SAP R/3 into Tivoli Workload Scheduler.
- “Importing factory calendars” on page 181 for an explanation about how you import the exported calendar definitions into the Tivoli Workload Scheduler database.

For details about the Tivoli Workload Scheduler calendar definitions, refer to the *Tivoli Workload Scheduler: User’s Guide and Reference*.

Exporting factory calendars

To export an SAP R/3 calendar, from *TWS_home*/methods (where *TWS_home* is the complete path where you installed Tivoli Workload Scheduler) enter the following command:

Command syntax

```
►►── -r3batch── -t RSC── -c──XAname── -- " -calendar_ID──calendarID──────────────►

                                        ┌─ -getworkdays─┐
►─ -year_from──yyyy── -year_to──yyyy──┤               ├────────────────────────►
                                        └─ -getfreedays─┘

►──┬───────────────────────────┬──┬─────────────────────────────────────┬─────►
   └─ -tws_name──tws_cal_name─┘  └─ -tws_description──tws_cal_desc─┘

►──┬──────────────────────────────┬─"──────────────────────────────────────►◄
   └─ -filename──output_filename─┘
```

Where:

-t RSC
: The identifier of the task to be performed, in this case RSC (Retrieve SAP R/3 Calendars). This parameter is required.

-c *XAname*
: The extended agent workstation connected to the SAP R/3 system where the calendar data to export is located. The SAP R/3 system must be configured as a workstation to Tivoli Workload Scheduler. This parameter is required.

-calendar_id *calendarID*
: The identifier of the SAP R/3 calendar to be exported, which consists of two alphanumeric characters. This parameter is required.

-year_from *yyyy*
The year of the calendar from when to start exporting dates, in the format *yyyy*. This parameter is required.

-year_to *yyyy*
The year of the calendar when to stop exporting dates, in the format *yyyy*. This parameter is required.

-getworkdays | -getfreedays
Specify getworkdays to create the Tivoli Workload Scheduler calendar definition based on the working days of the SAP R/3 calendar. In this way, each date of a working day is stored in the output file.

Specify getfreedays to create the Tivoli Workload Scheduler calendar definition based on the holidays of the SAP R/3 calendar. Each date of a non-working day is stored in the output file.

These parameters are optional and mutually exclusive. If you do not specify either, the default is getworkdays.

-tws_name *tws_cal_name*
The Tivoli Workload Scheduler name for the exported SAP R/3 factory calendar. It is stored in the output file.

You can specify up to eight alphanumeric characters. This parameter is optional, the default is SAP*XX*_*calendarID*, where:

XX Corresponds to WK if the calendar includes only working days or FR if the calendar includes only non-working days.

calendarID
The identifier of the SAP R/3 calendar.

For example, the default Tivoli Workload Scheduler name for an exported calendar, whose identifier is 04, that includes only working days, is SAPWK_04.

-tws_description *tws_cal_desc*
The description of the Tivoli Workload Scheduler calendar. It is stored in the output file. You can specify up to 120 alphanumeric characters. If the description contains blanks, it must be enclosed between single quotes. This parameter is optional.

-filename *output_filename*
The name of the output file that is to contain the calendar definitions. This file is written in a scheduling language that can be processed by the **composer** when you add the calendar data to the Tivoli Workload Scheduler database.

You can specify a file name with its complete or partial path; if you do not specify any path, the file is created in the current directory. If the path you specify does not exist, it is created, provided that you have the appropriate access rights. Otherwise, the command returns an error message and is not performed.

You can specify up to the maximum number of characters allowed by your operating system. If the name of the file contains blanks, it must be enclosed between single quotes. If another file with the same name exists, it is overwritten.

This parameter is optional. The default value is *tws_name*.txt, where *tws_name* is the value you set for the tws_name parameter.

The following is an example of an SAP R/3 factory calendar export command:

```
r3batch -t RSC -c tivoli10 -- " -calendar_id 01 -year_from 2007
-year_to 2010 -tws_name TWS_CAL1 -tws_description 'TWS SAP Calendar 01'
-getworkdays -filename 'my dir/tws_calendar_01.dat' "
```

This command exports the SAP R/3 calendar named 01, located on the SAP R/3 system named tivoli10. The dates exported begin from year 2007, until year 2010, considering only working days. The Tivoli Workload Scheduler name used for the calendar is TWS_CAL1, and the description written in the output file is TWS SAP Calendar 01. The output file is named `tws_calendar_01.dat`, stored in *`TWS_home`*`/methods/my dir`, and its content looks like the following:

```
$CALENDAR
    TWS_CAL1
    "TWS SAP Calendar 01"
    01/02/2007 01/03/2007 01/04/2007 01/05/2007 01/08/2007 01/09/2007 01/10/2007
    01/11/2007 01/12/2007 01/15/2007 01/16/2007 01/17/2007 01/18/2007 01/19/2007
    01/22/2007 01/23/2007 01/24/2007 01/25/2007 01/26/2007 01/29/2007 01/30/2007
    01/31/2007 02/01/2007 02/02/2007 02/05/2007 02/06/2007 02/07/2007 02/08/2007
     .......
    11/24/2010 11/25/2010 11/26/2010 11/29/2010 11/30/2010 12/01/2010 12/02/2010
    12/03/2010 12/06/2010 12/07/2010 12/08/2010 12/09/2010 12/10/2010 12/13/2010
    12/14/2010 12/15/2010 12/16/2010 12/17/2010 12/20/2010 12/21/2010 12/22/2010
    12/23/2010 12/24/2010 12/27/2010 12/28/2010 12/29/2010 12/30/2010 12/31/2010
```

Importing factory calendars

To import the exported calendar definitions into the Tivoli Workload Scheduler database, copy the output file from the extended agent for SAP R/3 to the master workstation and from the **composer** command line on the master workstation, enter the following command:

```
-add output_filename
```

where *`ouput_filename`* is the name of the exported file, with its complete path.

For example, to import the `tws_calendar_01.dat` file exported in the previous example, copy the file to the master workstation. From the **composer** command line on the master workstation, enter:

```
-add TWS_home/methods/my dir/tws_calendar_01.dat
```

where *`TWS_home`* is the complete path where you installed Tivoli Workload Scheduler.

Defining internetwork dependencies and event rules based on SAP R/3 background events

This chapter describes how to define internetwork dependencies and event rules for Tivoli Workload Scheduler based on SAP background events.

Note: To be able to define and monitor event rules, ensure that you configured your environment as described in “Configuring SAP event monitoring” on page 104.

Defining internetwork dependencies based on SAP background events

Dependencies are prerequisites that must be satisfied before a job or job stream can start. Internetwork dependencies are dependencies checked by the extended agent workstation to which they belong. In response to an internetwork dependency, the SAP extended agent checks for the occurrence of the SAP background event specified in the dependency. As soon as the SAP event is raised, the SAP extended agent commits the event and instructs Tivoli Workload Scheduler to resolve the corresponding internetwork dependency.

For more details about internetwork dependencies, refer to the *Tivoli Workload Scheduler: User's Guide and Reference*. For more details about how to raise SAP events, see "Raising an SAP event" on page 127.

To define SAP background events as internetwork dependencies, XBP versions 2.0 and 3.0 are supported, with the following differences:

XBP version 2.0

SAP background events can release Tivoli Workload Scheduler internetwork dependencies only if the dependencies are created or checked *before* the SAP event is raised. An event history is ignored, therefore an SAP event raised before the internetwork dependency is created, is not considered.

Note: Because an SAP event history is ignored, for each SAP background event to be checked, a placeholder SAP job is created. This is a dummy job whose running depends on the SAP background event, therefore an SAP event is considered raised as soon as the corresponding placeholder job has completed.

XBP version 3.0 (supported by SAP NetWeaver 7.0 with SP 9, or later)

Only the SAP background events stored in the SAP event history table are considered by Tivoli Workload Scheduler to check for internetwork dependencies resolution. As a prerequisite, the SAP administrator must create the appropriate event history profiles and criteria on the target SAP system.

To avoid performance reduction, run reorganization tasks against the SAP event history.

Note: Some SAP systems providing XBP version 3.0 still return XBP version as 2.0. To check if your SAP system provides XBP 3.0, invoke the transaction `se37` and search for the function module `BAPI_XBP_BTC_EVTHISTORY_GET`. If your system contains the module, set the `xbpversion` option to 3. In this way, **r3batch** will ignore the XBP value returned by the SAP system. For details about the `xbpversion` option, refer to Table 35 on page 95.

To define an SAP background event as an internetwork dependency, use the following parameters:

Table 45. Parameters to define an SAP internetwork dependency

Parameter	Description	GUI support
-evtid *sap_event_name*	The name of the SAP background event, up to 32 characters. If the name contains blanks, enclose it between single quotes. This parameter is required.	✓
-evtpar *sap_event_parm*	The SAP event parameter, up to 64 characters. If the parameter contains blanks, enclose it between single quotes. This parameter is optional.	✓
-commit	Defines that the SAP background event is committed immediately after the internetwork dependency has been resolved. If you do not specify `-commit`, the event must be committed by running the `r3batch` task PI. The default is that `-commit` is not specified. For details about the PI task, refer to "Committing SAP background events by an external task" on page 184. In addition to this parameter, you can set as default that the system commits internetwork dependencies immediately by specifying `commit_dependency=on` in the options file. For details about the commit_dependency option, see Table 35 on page 95. **Note:** With XBP version 2.0, defining two internetwork dependencies on the same SAP event might lead to an error, if `-commit` is specified. For example, suppose you define an internetwork dependency for the SAP event `SAPEVT`, with or without setting `-commit`. After this definition, the SAP event `SAPEVT` is raised. Then you define a second internetwork dependency based on `SAPEVT`, specifying `-commit`. The second dependency immediately commits the SAP event, with the consequence that the first dependency becomes impossible to resolve. Therefore, when the first job checks for the internetwork dependency, an error is issued.	✓

The following example shows how to define an internetwork dependency based on the SAP background event named `SAP_TEST` with the parameter `12345678`. After its processing, the event is not immediately committed.

```
-evtid SAP_TEST -evtpar 12345678
```

The resulting internetwork dependency looks like the following, where `SAPWS` is the name of the extended agent workstation that connects to the SAP background processing system where the event runs:

```
follows SAPWS::"-evtid SAP_TEST -evtpar 12345678"
```

The following example shows how to define an internetwork dependency based on the SAP background event named `SAP_TEST`, without parameter. As soon as the internetwork dependency is resolved, the event is committed.

```
-evtid SAP_TEST -commit
```

The resulting internetwork dependency looks like the following, where `SAPWS` is the name of the extended agent workstation that connects to the SAP background processing system where the event runs:

```
follows SAPWS::"-evtid SAP_TEST -evtpar 12345678"
```

Table 46 shows the correspondence between the definition and possible resolution of an internetwork dependency that depends on an SAP event, with or without parameters assigned. In this table, SAP_TEST is used as the event name and 12345678 or ABCDEFG as the event parameter.

Table 46. Internetwork dependency definition and possible resolution

Tivoli Workload Scheduler internetwork dependency specified	SAP event raised in SAP system	SAP event parameter	Tivoli Workload Scheduler internetwork dependency resolved
-evtid SAP_TEST	none	none	No
-evtid SAP_TEST	END_OF_JOB	none	No
-evtid SAP_TEST	SAP_TEST	none	Yes
-evtid SAP_TEST	SAP_TEST	12345678	Yes
-evtid SAP_TEST -evtpar 12345678	SAP_TEST	none	No
-evtid SAP_TEST -evtpar 12345678	SAP_TEST	12345678	Yes
-evtid SAP_TEST -evtpar 12345678	SAP_TEST	ABCDEFG	No

Committing SAP background events by an external task

SAP events defined as Tivoli Workload Scheduler internetwork dependencies, by default are not automatically committed after their processing. You can modify this default by specifying the -commit parameter. Otherwise, if you leave the default, you must commit the processed event by using the external task Put Information (PI).

The PI task commits all the processed events that meet the given criteria. For this reason, it is recommended that you run this task at the end of the working day. By doing so, internetwork dependencies that are already resolved are not reset and the objects depending on them are not blocked until they are resolved again.

From a command line, enter the following command:

Command syntax

```
►►── -r3batch── -t PI── -c──XAname── -- " -t CE── -evtid──sap_event_name──────────►

►──┬───────────────────────┬"──────────────────────────────────────────────────►◄
   └─ -evtpar──sap_event_parm─┘
```

Where:

-t PI The identifier of the task to be performed, in this case PI (Put Information). This parameter is required.

-c *XAname*
The extended agent workstation connected to the SAP background processing system where the event is run. This parameter is required.

-t CE The identifier of the task to be performed, in this case CE (Commit Event). This parameter is required.

-evtid *sap_event_name*
The name of the SAP R/3 event running on the background processing system. If the name contains blanks, enclose it between single quotes. This parameter is required.

-evtpar *sap_event_parm*
The parameter of the SAP event running on the background processing system. If the parameter contains blanks, enclose it between single quotes. This parameter is optional. If you do not specify it, all the SAP events with the name you specified, with or without a parameter, are committed on the target system.

The following is an example of how to commit the SAP event named `SAP_TEST`, with parameter `1234567`, running on the background processing system named `tivoli10`:

```
r3batch -t PI -c tivoli10 -- " -t CE -evtid SAP_TEST -evtpar 1234567"
```

Defining internetwork dependencies based on SAP background events with the Dynamic Workload Console

To define an SAP background event as an internetwork dependency with the Dynamic Workload Console, perform the following steps:

1. Launch the Workload Designer from the Dynamic Workload Consoleportfolio. Click **Workload** > **Design** > **Create Workload Definitions**.
2. Search for and open the job stream you want to manage.
   a. In the **Working List** pane, click **Search** > **Job Stream**.
   b. Type the job stream name or simply click **Search** to display all job streams.
   c. Select the job stream and click **Edit**. The job stream and its contents are displayed in the **Details** view.
3. Type the job stream name or simply click **Search** to display all job streams.
4. From the **Details** view, select the job or the job stream to which you want to add the dependency.
5. From the toolbar, click **Select an Action** > **Add Dependencies** > **Internetwork**.
6. Specify the properties for the internetwork dependency.
   a. In the **Network Agent** field, enter the name of the agent workstation connected to the SAP background processing system where the event runs.
   b. In the **Dependency** field, enter the parameters to define the internetwork dependency. For a description of the parameters allowed, refer to Table 45 on page 183.
7. Click **Save** to save the changes to the job stream.

The local job or job stream now has a dependency on a SAP background event. You can also perform this procedure from the graphical view available from the **Workload Designer**. For more information about adding dependencies and editing objects in the Graphical View, refer to the Dynamic Workload Console User’s Guide.

Defining event rules based on SAP background events

A scheduling event rule defines a set of actions to run when specific event conditions occur. The definition of an event rule correlates events and triggers actions.

An event rule is identified by a rule name and by a set of attributes that specify if the rule is active, the time frame of its validity, and other information required to decide when actions are triggered. It includes information related to the specific events (`eventCondition`) that the rule must detect and the specific actions it is to trigger upon their detection or timeout (`ruleAction`). Complex rules might include multiple events and multiple actions.

If you are using XBP 3.0, only the SAP background events that are stored in the event history table are considered by Tivoli Workload Scheduler.

To define event rules, you can use either of the following:

The composer command line
: You edit the rules with an XML editor of your choice. For details about how to use the composer to define event rules, see the *Tivoli Workload Scheduler User's Guide and Reference*.

The Dynamic Workload Console
: For information about creating an event rule, see Creating an event rule.For more details about the properties used to define the SAP event rule, see the following table available only in html format in the online information center: SAP Event Raised.

The SAP background event is identified by the following information:

SAP Event ID
: The name identifying the SAP event. Wildcards are not allowed.

If you are using the Dynamic Workload Console, you can type the event name in the SAP Event ID field. This field does not support wildcard characters (* and %), nor the following special characters: asterisk (*), question mark (?), and backslash (\). Note that for supported special characters, the escape character (\) must not be used.

Alternatively, you can use the lookup function to search for and select the event name. When specifying the string to search for that represents the SAP Event ID, wildcard characters are supported, (* and %). For example, if you specify "myevent*", then results can include events such as "myevent", "myevent%", and "myevents".

Event parameter
: The parameter associated with the SAP event, if any. Wildcards are not allowed.

If you are using the Dynamic Workload Console, the following special characters are not supported when specifying the event parameter: asterisk (*), question mark (?), and backslash (\).

Extended agent workstation
: The name of the extended agent workstation responsible for event monitoring.

Note:

1. If you specify a pattern with the wildcard asterisk (*), all the extended agents whose name matches the pattern will monitor the specified event.
2. As a best practice, define that an event belonging to an SAP system is monitored by one extended agent workstation only. If the same SAP event is monitored by more than one extended agent, you might either

be notified multiple times for the same event occurrence or the first extended agent that notifies the event occurrence makes that event unavailable to the other extended agents.

3. If you modify the extended agent configuration in the `r3batch` option files, to make the changes effective you must stop and restart the agent.

SAP events matching criteria

The SAP background events specified in the event rule are matched with the events raised in the SAP system, according to the following criteria. Depending on the parameters you set:

The SAP event ID and parameter are specified in the event rule

To match, the SAP event ID and parameter must be the same as the event ID and event parameter raised in the SAP system. Also, the event state must be N (New). SAP events with a different parameter or without any parameter are ignored.

The information collected about the matching SAP event is sent by the `r3evmon` process to Tivoli Workload Scheduler. If the notification is successfully sent, the event is committed on the SAP system and its state changed to C (Confirmed).

For example, you define an event rule in your Tivoli Workload Scheduler plan based on the following SAP event:

SAP event ID
SAP_TEST

SAP event parameter
ABCDEF

Extended agent workstation
GENIUS

According to these settings, a file named `GENIUS_r3evmon.cfg` is created on `GENIUS`. It contains the following `!R3EVENT` keyword:

```
!R3EVENT 0008SAP_TEST0006ABCDEF
```

Monitoring of the `SAP_TEST` event with parameter `ABCDEF` is automatically started. Suppose that the following SAP events were raised on the SAP system:

Table 47. History table of the SAP events raised

EVENT GUID	SAP EVENT ID	EVENT PARM	EVENT SERVER	EVENT TIMESTAMP	EVENT STATE	PROCESS STATE	COUNT OF JOBS
1234	SAP_TEST	ABC123	...	20070925 13:00	C	OK	1
2345	SAP_TEST	ABCD	...	20070925 14:00	N	OK	2
3456	SAP_TEST		...	20070925 15:00	N	OK	3
4567	SAP_TEST	ABCDEF	...	20070925 16:00	N	OK	4

Only the following SAP event is notified to Tivoli Workload Scheduler:

Table 48. SAP event matching with the event rule defined

EVENT GUID	SAP EVENT ID	EVENT PARM	EVENT SERVER	EVENT TIMESTAMP	EVENT STATE	PROCESS STATE	COUNT OF JOBS
4567	SAP_TEST	ABCDEF	...	20070925 16:00	N	OK	4

If the notification is successfully sent, the event is committed on the SAP system and its state changed to C (Confirmed).

Only the SAP event ID is specified in the event rule

To match, the SAP event ID must be the same as the ID of the events raised in the SAP system whose state is N (New). The parameters of the SAP events, whether specified or not, are not taken into account.

The information collected about all the matching SAP events is sent by the r3evmon process to Tivoli Workload Scheduler. Each event successfully notified is committed on the SAP system and its status changed to C (Confirmed).

For example, you define an event rule in your Tivoli Workload Scheduler plan based on the following SAP event:

SAP event ID
SAP_TEST

Extended agent workstation
GENIUS

According to these settings, a file named GENIUS_r3evmon.cfg is created on GENIUS. It contains the following !R3EVENT keyword:

```
!R3EVENT 0008SAP_TEST
```

Monitoring of the SAP_TEST event is automatically started. Suppose that the following SAP events were raised on the SAP system:

Table 49. History table of the SAP events raised

EVENT GUID	SAP EVENT ID	EVENT PARM	EVENT SERVER	EVENT TIMESTAMP	EVENT STATE	PROCESS STATE	COUNT OF JOBS
1234	SAP_TEST	ABC123	...	20070925 13:00	C	OK	1
2345	SAP_TEST	ABCD	...	20070925 14:00	N	OK	2
3456	SAP_TEST		...	20070925 15:00	N	OK	3
4567	SAP_TEST	ABCDEF	...	20070925 16:00	N	OK	4

Only the following SAP events are notified to Tivoli Workload Scheduler:

Table 50. SAP events matching with the event rule defined

EVENT GUID	SAP EVENT ID	EVENT PARM	EVENT SERVER	EVENT TIMESTAMP	EVENT STATE	PROCESS STATE	COUNT OF JOBS
2345	SAP_TEST	ABCD	...	20070925 14:00	N	OK	2
3456	SAP_TEST		...	20070925 15:00	N	OK	3
4567	SAP_TEST	ABCDEF	...	20070925 16:00	N	OK	4

Each event whose notification is successfully sent is committed on the SAP system and its state changed to C (Confirmed).

Setting a filter for SAP background events in the security file

In the security file, you can filter the SAP background events that can be used to define event rules. By doing this, you restrict the use of certain SAP events to specific users. For example, assume that you want your USA department to manage only the SAP events whose ID begins with SAP_USA, and your Italy department to manage all events except those beginning with SAP_USA. In the security file that defines the user access for the USA department, define the CUSTOM keyword for the EVENT object as follows:

```
EVENT      PROVIDER=@ +CUSTOM=SAP_USA@  ACCESS=USE
```

where:

PROVIDER=@
Specifies that the user can use the events coming from any provider.

+CUSTOM=SAP_USA@
Specifies that the user can use only the SAP events whose ID begins with SAP_USA.

This keyword applies only to the SAP provider (SapMonitor).

ACCESS=USE
Sets the user access to the object to USE.

In the security file that defines the user access for the Italy department, define the CUSTOM keyword for the EVENT object as follows:

```
EVENT      PROVIDER=@ ~CUSTOM=SAP_USA@  ACCESS=USE
```

where:

PROVIDER=@
Specifies that the user can use the events coming from any provider.

~CUSTOM=SAP_USA@
Specifies that the user can use all SAP events, except those whose ID begins with SAP_USA.

This keyword applies only to the SAP provider (SapMonitor).

ACCESS=USE
Sets the user access to the object to USE.

For more details about the security file and how to set up user authorizations, see the *Tivoli Workload Scheduler: User's Guide and Reference*.

Defining event rules based on IDoc records

With Tivoli Workload Scheduler for Applications, you can monitor Intermediate Document (IDoc) records in SAP systems and forward events to the Tivoli Workload Scheduler's event integration framework. You define an event condition that contains the criteria that the IDocs must match to be forwarded to Tivoli Workload Scheduler. When the event condition occurs, the action you associated with it (for example, running a job) is performed.

Business scenario

You connected your Internet sales application to your SAP Customer Relationship Management (CRM) system, which receives the orders as incoming IDocs. The orders are classified as emergency and ordinary, and therefore have different IDoc message types. You want the emergency orders to be imported into the CRM system directly, and the ordinary orders to be processed in batch mode. To do this, in Tivoli Workload Scheduler for Applications, you define an event rule that monitors the IDoc message types corresponding to emergency orders and sends an event to Tivoli Workload Scheduler. In Tivoli Workload Scheduler, you define a job to be released when this type of event is received and is linked to an SAP job that runs an import ABAP report for these specific types of IDocs.

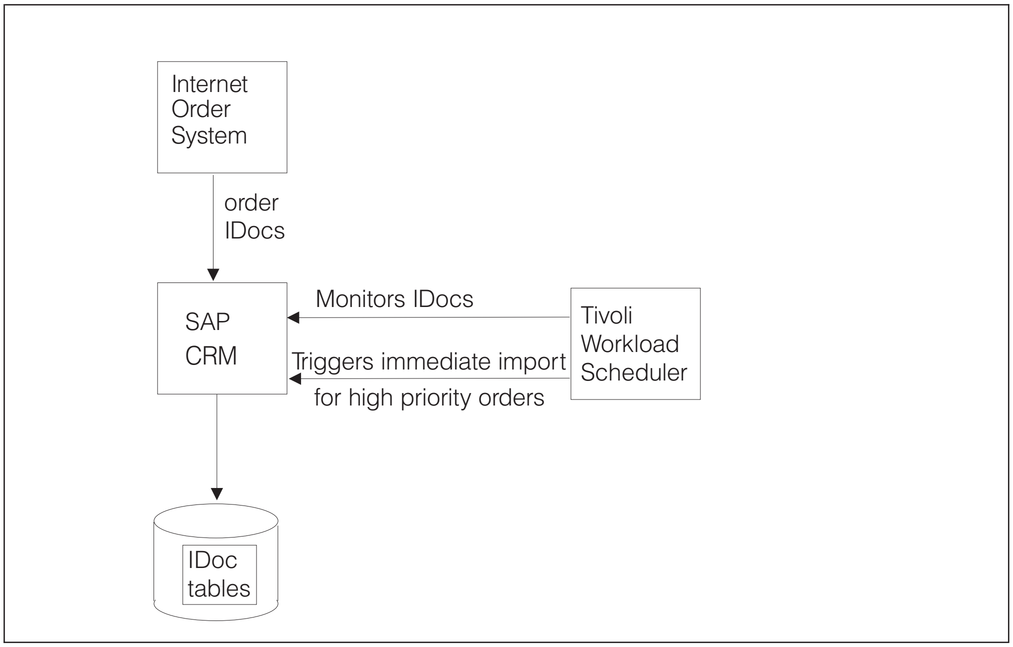


Figure 12. Managing high priority IDocs overview

Creating event rules based on IDocs

To define event rules based on IDocs, specify the fields to be used as matching criteria during IDoc monitoring. For details about these fields, refer to "Events matching criteria" on page 191. To create the event rules, you can use either of the following:

The composer command line

You edit the rules with an XML editor of your choice. For a general explanation about how to use the composer to define event rules, see the *Tivoli Workload Scheduler: User's Guide and Reference*. The event condition requires:

- `SAPMonitor` as event monitor provider.
- `IDOCEventGenerated` as event type.

For a list of the values that you can specify in the **attributeFilter** name when defining the event condition, refer to Table 53 on page 192.

The Dynamic Workload Console

| For information about creating an event rule, see Creating an event
| rule.For more details about the properties used to define the IDoc event
| rule, see the following table available only in html format in the online
| information center: Event Raised.

Note:

1. To be able to define and monitor event rules, ensure that you configured your environment as described in "Configuring SAP event monitoring" on page 104.
2. To configure how Tivoli Workload Scheduler for Applications retrieves the IDoc monitors, set **idoc_no_history** and **idoc_shallow_result** in the options file. For details about these options, refer to "Defining the common options" on page 94.

Events matching criteria

Table 51 lists the Tivoli Workload Scheduler fields corresponding to the fields in the IDoc record that you want to search. During monitoring, each IDoc matching the search criteria generates an event that is sent to Tivoli Workload Scheduler.

Table 51. Tivoli Workload Scheduler fields used to define event rules based on IDocs

Composer property	Console property	IDoc field
SAPClient	SAP client	MANDT
SAPIDocStatus	Status	STATUS
SAPDirectionIDocTransmission	Direction	DIRECT
SAPReceiverPort	Receiver port	RCVPOR
SAPReceiverPartnerFunction	Receiver partner function	RCVPFC
SAPReceiverPartnerType	Receiver partner type	RCVPRT
SAPReceiverPartnerNumber	Receiver partner number	RCVPRN
SAPSenderPort	Sender port	SNDPOR
SAPSenderPartnerType	Sender partner type	SNDPRT
SAPSenderPartnerFunction	Sender partner function	SNDPFC
SAPSenderPartnerNumber	Sender partner number	SNDPRN
SAPLogicalMessageType	Logical message type	MESTYP
SAPNameOfBasicType	Name of basic type	IDOCTP
SAPLogicalMessageCode	Logical message code	MESCOD
SAPLogicalMessageFunction	Logical message function	MESFCT
SAPTestFlag	Test flag	TEST
SAPOutputMode	Output mode	OUTMOD

Optionally, you can define also correlation rules by using the fields listed in Table 52. Date and time values are specified in GMT time zone.

Table 52. Tivoli Workload Scheduler fields used to define correlation rules for IDoc events

Composer property	Console property	IDoc field
SAPIDocNumber	IDoc number	DOCNUM
SAPReleaseForIDoc	IDoc SAP release	DOCREL
SAPIDocType	IDoc type	DOCTYP
SAPReceiverAddress	Receiver SADR address	RCVSAD
SAPReceiverSADRClient	Receiver SADR client	RCVSMN
SAPFlagForInternationalReceiverAddress	Receiver SADR flag	RCVSNA
SAPReceiverCommunicationType	Receiver SADR communication type	RCVSCA
SAPDefaultFlagForReceiverAddress	Receiver SADR default flag	RCVSDF
SAPReceiverAddressSequentialNumber	Receiver SADR sequential number	RCVSLF
SAPReceiverLogicalAddress	Receiver logical address	RCVLAD
SAPEDIStandard	EDI Standard	STD
SAPEDIStandardVersion	EDI standard version	STDVRS
SAPEDIMessageType	EDI message type	STDMES

Table 52. Tivoli Workload Scheduler fields used to define correlation rules for IDoc events (continued)

Composer property	Console property	IDoc field
SAPSenderAddress	Sender SADR address	SNDSAD
SAPSenderSADRClient	Sender SADR client	SNDSMN
SAPFlagForInternationalSenderAddress	Sender SADR flag	SNDSNA
SAPSenderCommunicationType	Sender SADR communication type	SNDSCA
SAPDefaultFlagForSenderAddress	Sender SADR default flag	SNDSDF
SAPSenderAddressSequentialNumber	Sender SADR sequential number	SNDSLF
SAPSenderLogicalAddress	Sender logical address	SNDLAD
SAPReferenceToInterchangeFile	Interchange file reference	REFINT
SAPReferenceToMessageGroup	Message group reference	REFGRP
SAPReferenceToMessage	Message reference	REFMES
SAPEDIArchiveKey	EDI archive key	ARCKEY
SAPIDocCreationDate	IDoc creation date	CREDAT
SAPIDocCreationTime	IDoc creation time	CRETIM
SAPExtension	Extension	CIMTYP
SAPEDIALESerializationField	EDI/ALE Serialization field	SERIAL
SAPOverridingInInboundProcessing	Overriding in inbound processing	EXPRSS
SAPIDocChangeDate	IDoc last update date	UPDDAT
SAPIDocChangeTime	IDoc last update time	UPDTIM

Based on the defined rule, the r3evmon process of Tivoli Workload Scheduler for Applications monitors the events related to IDoc records according to a polling rate. To customize this polling rate, use the **evmon_interval** option; for details, see "Defining the common options" on page 94.

Table 53 lists the values that you can specify as attribute filter name when defining the event condition.

Table 53. Parameters of IDOCEventGenerated event type

Property name	Description	Type	Filtering allowed	Required	Multiple values allowed	Wildcard allowed	Length (min-max)	
SAPClient	SAP client number	numeric (0-9)	✓	✓		✓	1	3
SAPIDocStatus	IDoc status information For a list of allowed values, refer to Table 54 on page 193 and Table 55 on page 194.	numeric	✓	✓	✓		1	2
SAPDirectionIDocTransmission	IDoc direction	numeric Value can be 1 (outbound) or 2 (inbound).	✓	✓			1	1
SAPReceiverPort	Receiver port. SAP system, EDI subsystem	string	✓				1	10

Table 53. Parameters of IDOCEventGenerated event type (continued)

Property name	Description	Type	Filtering allowed	Required	Multiple values allowed	Wildcard allowed	Length (min-max)	
SAPReceiverPartnerFunction	Partner function of receiver	string	✓				1	2
SAPReceiverPartnerType	Partner type of receiver	string	✓				1	2
SAPReceiverPartnerNumber	Partner number of receiver	string	✓				1	10
SAPSenderPort	Sender port. SAP system, EDI subsystem	string	✓				1	10
SAPSenderPartnerType	Partner type of sender	string	✓				1	2
SAPSenderPartnerFunction	Partner function of sender	string	✓				1	2
SAPSenderPartnerNumber	Partner number of sender	string	✓				1	10
SAPLogicalMessageType	Logical message type	string	✓			✓	1	30
SAPNameOfBasicType	Name of basic type	string	✓			✓	1	30
SAPLogicalMessageCode	Logical message code	string	✓				1	3
SAPLogicalMessageFunction	Logical message function	string	✓				1	3
SAPTestFlag	Test flag	string	✓				1	1
SAPOutputMode	Output Mode	string Value can be 2 (immediate sending) or 4 (collected sending).	✓				1	1

Table 54 lists the standard outbound IDoc statuses and Table 55 on page 194 lists the standard inbound IDoc statuses. Optionally, you can activate a check to prevent event rule definitions with inconsistent IDoc status list and direction. If you activate the check and specify inconsistent values when defining a rule (for example, 02 as status and 2 as direction), you receive an error message and you cannot save the rule definition. To activate the check, perform the following steps:

1. In the `TWS_home/eventPlugIn` directory, create the `SapMonitorPlugIn.properties` file.
2. Edit `SapMonitorPlugIn.properties` to set the following configuration property:
   ```
   TWSPlugIn.event.idoc.consistency.check = true
   ```
3. From conman, stop and restart the event processing server by using, respectively, the **stopeventprocessor** and **starteventprocessor** commands.

The default value is `false`.

To have predictable event action results, when defining event rules consider using only non-transitory statuses that allow user checks.

Table 54. Standard outbound IDoc statuses

Status	Description
01	IDoc generated
02	Error passing data to port
03	Data passed to port
04	Error within control information of EDI subsystem
05	Error during translation

Table 54. Standard outbound IDoc statuses (continued)

Status	Description
06	Translation
07	Error during syntax check
08	Syntax check
09	Error during interchange
10	Interchange handling
11	Error during dispatch
12	Dispatch OK
13	Retransmission OK
14	Interchange acknowledgement positive
15	Interchange acknowledgement negative
16	Functional acknowledgement positive
17	Functional acknowledgement negative
18	Triggering EDI subsystem OK
19	Data transfer for test OK
20	Error triggering EDI subsystem
22	Dispatch OK, acknowledgement still due
23	Error during retransmission
24	Control information of EDI subsystem OK
25	Processing despite syntax error
26	Error during syntax check of IDoc
27	Error in dispatch level (ALE service)
29	Error in ALE service
30	IDoc ready for dispatch (ALE service)
31	Error no further processing
32	IDoc was edited
33	Original of an IDoc which was edited
34	Error in control record of IDoc
36	Electronic signature not performed (timeout)
37	IDoc added incorrectly
38	IDoc archived
39	IDoc is in the target system (ALE service)
40	Application document not created in target system
41	Application document created in target system
42	IDoc was created by test transaction

Table 55. Standard inbound IDoc statuses

Status	Description
50	IDoc added
51	Application document not posted
52	Application document not fully posted

Table 55. Standard inbound IDoc statuses (continued)

Status	Description
53	Application document posted
54	Error during formal application check
55	Formal application check OK
56	IDoc with errors added
57	Error during application check
58	IDoc copy from R/2 connection
60	Error during syntax check of IDoc
61	Processing despite syntax error
62	IDoc passed to application
63	Error passing IDoc to application
64	IDoc ready to be transferred to application
65	Error in ALE service
66	IDoc is waiting for predecessor IDoc (serialization)
68	Error - no further processing
69	IDoc was edited
70	Original of an IDoc which was edited
71	IDoc reloaded from archive
73	IDoc archived
74	IDoc was created by test transaction

For example, you define a rule with the following attributes:

Extended agent workstation
SAPCPU

SAP client number
001

IDoc status list
56,60

IDoc direction
2 (inbound)

After saving the rule according to these settings, when the rule becomes active a file named SAPCPU_r3evmon.cfg is created on SAPCPU. It contains the following !IDOC keyword:

```
!IDOC 0003001000556,600001200000000000000000000000000000000000000000000000000000000
```

IDoc monitoring is automatically started. When the event condition is verified, the action defined in the rule is triggered

For an explanation of the !IDOC keyword format, refer to Table 88 on page 286.

Examples of event rules based on IDocs

The following example applies to the scenario described in “Business scenario” on page 189. It shows an event rule that triggers an import ABAP report when an IDoc is added with a message type corresponding to emergency orders.

```
<?xml version="1.0" encoding="UTF-8"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules"
              xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/
                  event-management/rules EventRules.xsd">
        <eventRule name="scenario1_IDoc" ruleType="filter" isDraft="no">
          <eventCondition name="IDocEventRaised1" eventProvider="SapMonitor"
                          eventType="IDocEventGenerated">
             <scope>
                   001 ON SAPCU WITH 2
             <scope>
             <filteringPredicate> <attributeFilter name="Workstation" operator="eq">
                    <value>SAPCPU</value>
                </attributeFilter>
                <attributeFilter name="SAPClient" operator="eq">
                    <value>001</value>
                </attributeFilter>
                <attributeFilter name="SAPIDocStatus" operator="eq">
                    <value>50</value>
                </attributeFilter>
                <attributeFilter name="SAPDirectionIDocTransmission" operator="eq">
                    <value>2</value>
                </attributeFilter>
                <attributeFilter name="SAPLogicalMessageType" operator="eq">
                    value>EORD1</value>
                </attributeFilter>
            </filteringPredicate>
          </eventCondition>
          <action actionProvider="TWSaction" actionType="sbj"
                  responseType="onDetection">
            <description>Trigger immediate report for high priority orders
            </description>
            <parameter name="JobDefinitionWorkstationName">
              <value>MASTER84</value>
            </parameter>
            <parameter name="JobDefinitionName">
              <value>triggerimport</value>
            </parameter>
          </action>
        </eventRule>
</eventRuleSet>
```

The following example shows an event rule defined to create a ticket for failing IDocs in the SAP Solution Manager or any other problem management system: when an IDoc with a syntax error is detected, the engine submits a job to create a ticket for the failing IDoc.

```
<?xml version="1.0" encoding="UTF-8"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules"
              xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/
                  event-management/rules EventRules.xsd">
        <eventRule name="scenario1_IDoc" ruleType="filter" isDraft="no">
          <eventCondition name="IDocEventRaised1" eventProvider="SapMonitor"
                          eventType="IDocEventGenerated">
             <filteringPredicate> <attributeFilter name="Workstation" operator="eq">
                    <value>SAPCPU</value>
                </attributeFilter>
                <attributeFilter name="SAPClient" operator="eq">
                    <value>001</value>
                </attributeFilter>
                <attributeFilter name="SAPIDocStatus" operator="eq">
                    <value>60</value>
                <attributeFilter name="SAPDirectionIDocTransmission" operator="eq">
                    <value>2</value>
                </attributeFilter>
                </attributeFilter>
                <attributeFilter name="SAPLogicalMessageType" operator="eq">
                    value>MYORD1</value>
                </attributeFilter>
            </filteringPredicate>
          </eventCondition>
          <action actionProvider="TWSaction" actionType="sbj"
                  responseType="onDetection">
            <description>Create a ticket for failing IDocs
            </description>
            <parameter name="JobDefinitionWorkstationName">
              <value>MASTER84</value>
            </parameter>
            <parameter name="JobDefinitionName">
              <value>createticket</value>
            </parameter>
          </action>
        </eventRule>
</eventRuleSet>
```

Defining event rules based on CCMS Monitoring Architecture alerts

The SAP Computing Center Monitoring System (CCMS) is a centralized monitoring architecture that provides a set of monitors for monitoring the SAP environment. Using the CCMS functions you can check the performance of the various SAP system components, diagnose potential problems, and be alerted about error and warning conditions. The monitors provide you with the information you require to fine tune the SAP system and the operating modes, and hence optimize system performance.

With Tivoli Workload Scheduler for Applications, you can integrate the CCMS monitoring functions into your management infrastructure by defining event rules based on the alerts raised in the SAP system.

Business scenarios

The following sections describe:

- "Business scenario: defining an event rule to process alerts related to IDocs"
- "Business scenario: defining an event rule to process alerts related to operating system" on page 198

Business scenario: defining an event rule to process alerts related to IDocs

You connected your Internet sales application to your SAP Customer Relationship Management (CRM) system, which receives the orders as incoming IDocs. You want to import the orders into the CRM system when their number exceeds a specified threshold, therefore you configured your SAP CCMS monitoring architecture to generate an alert when the number of incoming IDocs exceeds a certain value. To automatically start a task that imports the orders:

1. In your SAP CCMS monitoring architecture, identify the element related to the alert that you configured for the incoming order IDocs.
2. In Tivoli Workload Scheduler, define an event rule, to be active during the timeframe when inbound order traffic is heavy, which monitors the element identified in step 1. As soon as an alert is generated for the element, a CCMS event is sent to Tivoli Workload Scheduler.
3. In Tivoli Workload Scheduler, define a job to be submitted when the CCMS event is received, to run an SAP job that runs an import ABAP report for the order IDocs.

Business scenario: defining an event rule to process alerts related to operating system

As a Tivoli Workload Scheduler for Applications administrator, you are in charge of taking the appropriate action in the Tivoli Workload Scheduler plan when a critical situation occurs in the SAP system. You have an SAP extended agent workstation dedicated to submit Business Intelligence tasks, whose activity you want to suspend every time the SAP BI system faces a critical situation (for example, the SAP system is running out of space). To do this:

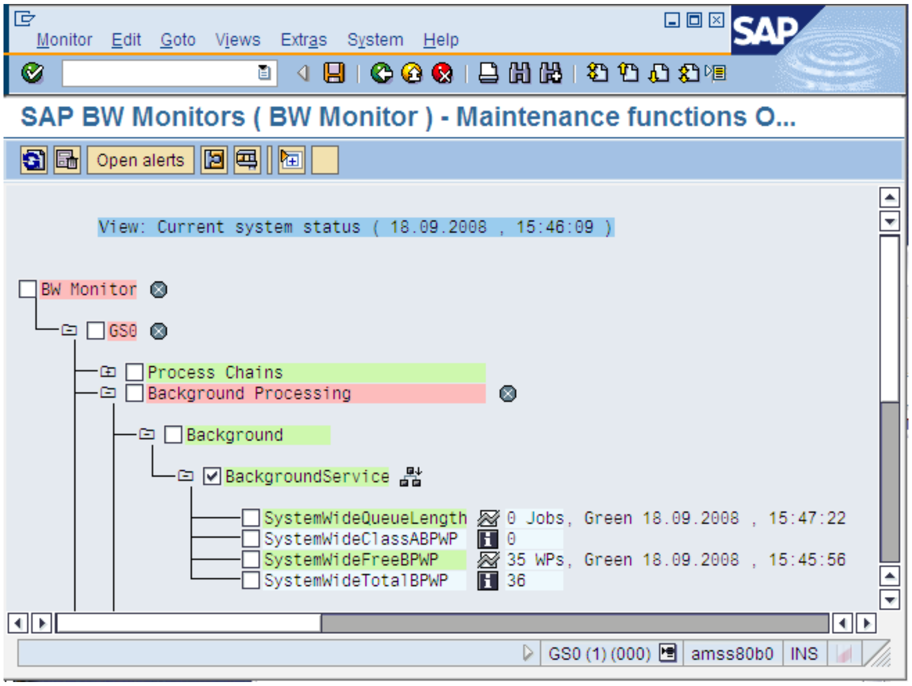
1. In your SAP CCMS monitoring architecture, identify the element related to the SAP system you want to monitor.
2. In Tivoli Workload Scheduler, define an event rule that monitors the element and sends an event to Tivoli Workload Scheduler when an alert is generated for it. Associate with this event an action that sets the limit of the extended agent workstation to 0, and sends a mail to the SAP administrator to notify the details of the critical situation.
3. As soon as the SAP administrator resolves the problem, you set the limit of the extended agent workstation back to its original value to resume the scheduling activities.

Creating event rules based on CCMS alerts

SAP systems are shipped with a predefined set of monitors, grouped in *monitor sets*. A monitor set contains a list of monitors, each monitor contains a set of *monitoring trees*. A monitor is a set of *monitoring tree elements* (MTEs) that are arranged in a hierarchical structure, named *alert monitoring tree*. You can define event rules based on the alert generated for a specific MTE.

Note: To be able to define and monitor event rules, ensure that you configured your environment as described in “Configuring SAP event monitoring” on page 104.

Figure 13 on page 199 shows the monitor named **BW Monitor** (belonging to the monitor set **SAP BW Monitor**) and its associated monitor tree elements (MTEs).



```
View: Current system status ( 18.09.2008 , 15:46:09 )

□ BW Monitor
  □ GS0
    □ Process Chains
    □ Background Processing
      □ Background
        ☑ BackgroundService
          □ SystemWideQueueLength   0 Jobs, Green 18.09.2008 , 15:47:22
          □ SystemWideClassABPWP    0
          □ SystemWideFreeBPWP      35 WPs, Green 18.09.2008 , 15:45:56
          □ SystemWideTotalBPWP     36
```

Figure 13. A monitor and its MTEs - © SAP AG 2009. All rights reserved.

To configure how Tivoli Workload Scheduler for Applications retrieves the CCMS alerts, set **ccms_alert_history** in the options file. For details about this option, refer to “Defining the common options” on page 94.

To create event rules, you can use either of the following:

The composer command line

You edit the rules with an XML editor of your choice. For a general explanation about how to use the composer to define event rules, see the *Tivoli Workload Scheduler User's Guide and Reference*.

The Dynamic Workload Console

For information about creating an event rule, see Creating an event rule.For more details about the properties used to define the CCMS event rule, see the following table available only in html format in the online information center: SAP Event Raised.

To define the CCMS event for your rule, specify the following information. For more details about how you separate the MTE name into the individual Tivoli Workload Scheduler fields, see “Mapping between the MTE name and Tivoli Workload Scheduler fields” on page 200.

XA Workstation

Name of the extended agent workstation responsible for the alert monitoring. This field is required. Wildcards are allowed.

MTE SAP System ID

Name of the SAP system where the MTE is located (for example, GS0 in Figure 13). This field is required. Wildcards are not allowed, you can specify up to eight characters.

MTE Monitoring Context Name

Name of the monitoring context to which the MTE belongs. This field is required. A monitoring context is a logically connected group of monitoring objects that are ordered together under one summary in the monitoring tree (for example, `Background` in Figure 13 on page 199).

Wildcards are not allowed, you can specify up to 40 characters.

MTE Monitoring Object Name

Name of the monitoring object in the alert monitor. This field is required. A monitoring object is a component or property of the system that is to be monitored (for example, `BackgroundService` in Figure 13 on page 199). If you choose not to specify a value, you must leave the value NULL, which is the default.

Wildcards are not allowed, you can specify up to 40 characters.

MTE Monitoring Attribute Name

Name of the monitoring attribute in the alert monitor. In the monitoring tree, a monitoring attribute is always an end node in the hierarchy (for example, `SystemWideFreeBPWP` in Figure 13 on page 199). This field is required. If you choose not to specify a value, you must leave the value NULL, which is the default.

Wildcards are not allowed, you can specify up to 40 characters.

Alert Value

Numeric value that indicates the color of the alert generated for the MTE. This field is optional. You can specify one or a combination of the following values:

1 Green, meaning Everything OK.
2 Yellow, meaning Warning.
3 Red, meaning Problem or error.

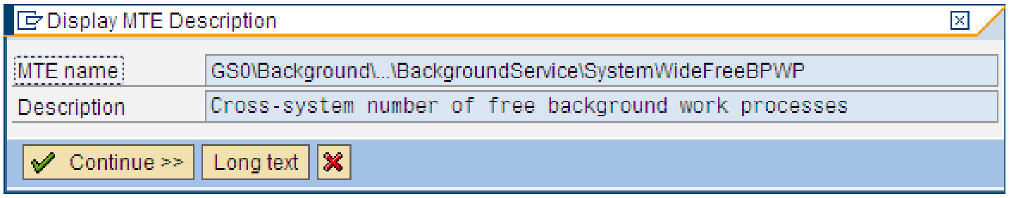
If you do not specify any value, all the alerts generated for the MTE are considered.

Alert Severity

Severity of the alert. It can be a number between 0 (lowest) and 255 (highest), or a range among these values. This field is optional. Alert severity is assigned during alert configuration; the SAP standard configuration is 50.

Mapping between the MTE name and Tivoli Workload Scheduler fields

Within SAP, MTEs are identified by a name made up of several tokens, separated by backslashes (\). To display the complete MTE name, select the MTE and click **Properties** or press **F1**:



MTE name	GS0\Background\...\BackgroundService\SystemWideFreeBPWP
Description	Cross-system number of free background work processes

Figure 14. Name and description of an MTE - © SAP AG 2009. All rights reserved.

According to the type of MTE that you want to monitor, you must fill in each Tivoli Workload Scheduler field with a specific token of the MTE name (to know your MTE type, select the MTE and click **Legend**):

If you are using Dynamic Workload Console V8.5.1, or later

1. In the **Event Rule Editor** panel, from the **Properties** section, click **Autofill MTE Tokens**. The **MTE Name** window opens.
2. In the **MTE Name** field, write the name of the MTE to monitor and click **OK**. You are returned to the **Event Rule Editor** panel, where the Tivoli Workload Scheduler fields are filled in accordingly.

If you are using Dynamic Workload Console prior to V8.5.1

Refer to the instructions provided in the following sections:

- "Context MTE"
- "Object MTE" on page 202
- "Attribute MTE" on page 202

Note: Virtual MTEs cannot be monitored.

Context MTE: A context MTE is the uppermost node of a monitoring tree; it contains all the associated object MTEs and attribute MTEs. Context nodes can be either of the following types:

Root Belongs only to the All Monitoring Contexts monitor. According to the SAP version you are using, a root context MTE name can have either of the following formats:

```
tokenA\tokenB\...
- OR -
tokenA\tokenB
```

For example:

```
T10\SystemConfiguration\...
```

Refer to Table 56 for an explanation about how you report this type of MTE in the Tivoli Workload Scheduler fields:

Table 56. Mapping between root context MTE name and Tivoli Workload Scheduler fields

Tivoli Workload Scheduler field	Token of MTE name	In this example...
MTE SAP System ID	*tokenA*	`T10`
MTE Monitoring Context Name	*tokenB*	`SystemConfiguration`
MTE Monitoring Object Name	N/A	`NULL`
MTE Monitoring Attribute Name	N/A	`NULL`

Summary

According to the SAP version you are using, a summary context MTE name can have either of the following formats:

```
tokenA\tokenB\...\tokenC\...
- OR -
tokenA\tokenB\tokenC
```

For example:

```
T10\SystemConfiguration\...\InstalledSupportPackages\...
```

Refer to Table 57 on page 202 for an explanation about how you report this type of MTE in the Tivoli Workload Scheduler fields:

Table 57. Mapping between summary context MTE name and Tivoli Workload Scheduler fields

Tivoli Workload Scheduler field	Token of MTE name	In this example...
MTE SAP System ID	*tokenA*	`T10`
MTE Monitoring Context Name	*tokenB*	`SystemConfiguration`
MTE Monitoring Object Name	*tokenC*	`InstalledSupportPackages`
MTE Monitoring Attribute Name	N/A	`NULL`

Object MTE: According to the SAP version you are using, an object MTE name can have either of the following formats:

```
tokenA\tokenB\tokenC\tokenD
- OR -
tokenA\tokenB\...\tokenD
```

For example:

```
PR0\amsp53_PR0_11\R3Services\Background\
```

Refer to Table 58 for an explanation about how you report this type of MTE in the Tivoli Workload Scheduler fields:

Table 58. Mapping between object MTE name and Tivoli Workload Scheduler fields

Tivoli Workload Scheduler field	Token of MTE name	In this example...
MTE SAP System ID	*tokenA*	`PR0`
MTE Monitoring Context Name	*tokenB*	`amsp53_PR0_11`
MTE Monitoring Object Name	*tokenD*	`Background`
MTE Monitoring Attribute Name	N/A	`NULL`

Attribute MTE: According to the SAP version you are using, an attribute MTE name can have either of the following formats:

```
tokenA\tokenB\tokenC\tokenD\tokenE
- OR -
tokenA\tokenB\...\tokenD\tokenE
```

For example:

```
PR0\amsp53_PR0_11\R3Services\Background\AbortedJobs
```

Refer to Table 59 for an explanation about how you report this type of MTE in the Tivoli Workload Scheduler fields:

Table 59. Mapping between attribute MTE name and Tivoli Workload Scheduler fields

Tivoli Workload Scheduler field	Token of MTE name	In this example...
MTE SAP System ID	*tokenA*	`PR0`
MTE Monitoring Context Name	*tokenB*	`amsp53_PR0_11`
MTE Monitoring Object Name	*tokenD*	`Background`
MTE Monitoring Attribute Name	*tokenE*	`AbortedJobs`

Setting correlation rules and action parameters

Optionally, you can use the alert properties listed in Table 60 on page 203 to:

- Define correlation rules between CCMS events.

- Specify additional parameters for the action that is associated with the event rule.

Date and time values are specified in GMT time zone.

Table 60. Alert properties for correlations

CCMS alert property	Console property	Composer property
MTSYSID for event definition	MTE SAP System ID	InputSAPSystemID
MTMCNAME for event definition	MTE Monitoring Context Name	InputMonitoringContextName
OBJECTNAME for event definition	MTE Monitoring Object Name	InputMonObjectName
FIELDNAME for event definition	MTE Monitoring Attribute Name	InputMonFieldName
ALERTDATE	Alert Date	AlertDate
ALERTTIME	Alert Time	AlertTime
ALINDEX	Alert Index	AlertIndex
ALSYSID	Alert SAP System ID	AlertSAPSystemID
ALUNIQNUM	Alert UID	AlertUID
EXTINDEX	Alert MT Ext Index	AlertMTExtIndex
FIELDNAME	Alert Monitoring Attribute Name	AlertMonFieldName
MANDT	Client	Client
MSCGLID	Alert Msg Container ID	AlertMsgContainerID
MSEGNAME	Alert MTE Segment	AlertMTESegment
MSG	Alert Message	AlertMessage
MSGCLASS	XMI Ext Company Name	XMIExtCompanyName
MSGID	XMI Log Msg ID	XMILogMsgID
MTCLASS	Alert MT Class	AlertMTClass
MTINDEX	Alert MT Index	AlertMTIndex
MTMCNAME	Alert Monitoring Context Name	AlertMTEContext
MTNUMRANGE	Alert MTE Range	AlertMTERange
MTSYSID	Alert MTE System	AlertMTESys
MTUID	Alert MT Type ID	AlertMTTypeID
OBJECTNAME	Alert Monitoring Object Name	AlertMonObjName
RC	Alert Return Code	AlertReturnCode
REPORTEDBY	Alert Reported By	AlertReportedBy
SEVERITY	Alert Severity	AlertSeverity
STATCHGBY	Alert Changed By	AlertChangedBy
STATCHGDAT	Alert Change Date	AlertChangeDate
STATCHGTIM	Alert Change Time	AlertChangeTime
STATUS	Alert Status	AlertStatus
USERID	User ID	UserID

Table 60. Alert properties for correlations (continued)

CCMS alert property	Console property	Composer property
VALUE	Alert Value	AlertValue

Getting alert status and committing alerts by an external task

Learn how to get CCMS alert status and commit CCMS alerts.

Refer to the following sections for details about:

- "Getting CCMS alert status"
- "Committing CCMS alerts" on page 205

Getting CCMS alert status

To get the current status of a CCMS alert from Tivoli Workload Scheduler, use the external task Get Information (GI). To replace the command arguments with the actual values, refer to the output returned by the event rule you defined. For details about the correspondence between the CCMS properties and the Console and composer properties, see Table 60 on page 203.

From a command line, enter the following command:

Command syntax

```
►►── -r3batch── -t GI── -c──XAname── -- " -t GAS── -alsysid──sap_system_ID────────►

►─ -msegname──alert_mte_segment── -aluniqnum──alert_UID───────────────────────────►

►─ -alindex──alert_index── -alertdate──alert_date── -alerttime──alert_time──"────►◄
```

Where:

-t GI Identifier of the task to be performed, in this case GI (Get Information).

-c *XAname*
Name of the extended agent workstation connected to the SAP system where the MTE for which the alert was raised is located.

-t GAS
Identifier of the task to be performed, in this case GAS (Get Alert Status).

-alsysid *sap_system_ID*
Identifier of the SAP system where the MTE for which the alert was raised is located. If the name contains blanks, enclose it between single quotes.

-msegname *alert_monitoring_segment*
Name of the alert monitoring segment. You can specify from 1 to 40 characters.

-aluniqnum *alert_UID*
Unique identifier of the alert, made up of 10 characters.

-alindex *alert_index*
Alert index, made up of 10 characters.

-alertdate *alert_date*
Date of the alert, in the format *yyyymmdd*.

-alerttime *alert_time*
Time of the alert, in the format *hhmmss*.

The following is an example of how to retrieve the current status of a CCMS alert:

```
r3batch -t GI -c tivoli10 -- " -t GAS -alsysid T10
        -msegname SAP_CCMS_tivoli10_T10_00 -aluniqnum 0017780869
        -alindex 0000000104 -alertdate 20081007 -alerttime 040356"
```

You are returned the current status of the alert.

Committing CCMS alerts

The CCMS alerts that you defined as Tivoli Workload Scheduler events are not automatically committed after their processing. To commit an alert that was processed by Tivoli Workload Scheduler, use the external task Put Information (PI).

To replace the command arguments with the actual values, refer to the output returned by the event rule you defined. For details about the correspondence between the CCMS properties and the Console and composer properties, see Table 60 on page 203.

From a command line, enter the following command:

Command syntax

```
►►── -r3batch── -t PI── -c──XAname── -- " -t CA── -alsysid──sap_system_ID──────────►

►─ -msegname──alert_mte_segment── -aluniqnum──alert_UID─────────────────────────────►

►─ -alindex──alert_index── -alertdate──alert_date── -alerttime──alert_time──"─────►◄
```

Where:

-t PI Identifier of the task to be performed, in this case PI (Put Information).

-c *XAname*
Name of the extended agent workstation connected to the SAP system where the MTE for which the alert was raised is located.

-t CA Identifier of the task to be performed, in this case CA (Commit Alert).

-alsysid *sap_system_ID*
Identifier of the SAP system where the MTE for which the alert was raised is located. If the name contains blanks, enclose it between single quotes.

-msegname *alert_monitoring_segment*
Name of the alert monitoring segment. You can specify from 1 to 40 characters.

-aluniqnum *alert_UID*
Unique identifier of the alert, made up of 10 characters.

-alindex *alert_index*
Alert index, made up of 10 characters.

-alertdate *alert_date*
Date of the alert, in the format *yyyymmdd*.

-alerttime *alert_time*
Time of the alert, in the format *hhmmss*.

The following is an example of how to commit a CCMS alert:

```
r3batch -t PI -c tivoli10 -- " -t CA -alsysid T10
        -msegname SAP_CCMS_tivoli10_T10_00 -aluniqnum 0017780869
        -alindex 0000000104 -alertdate 20081007 -alerttime 040356"
```

You are returned with the message `The CCMS alert was successfully confirmed`.

Example of an event rule based on CCMS alerts

The following example shows an event rule defined to monitor the yellow alerts raised on the MTE named `GS0\ALE/EDI GS0(000) Log.sys TVALE\Inbound IDoc ORDER_IDOC\Inbound: IDoc generated`. The MTE is configured to generate a yellow alert when the number of IDocs representing orders ready to process exceeds a specified threshold. If this condition occurs, the following actions are triggered:

- A Tivoli Workload Scheduler for Applications job is submitted to process the order IDocs.
- A Tivoli Workload Scheduler job, with priority 10, is submitted to confirm the alert.

```
<?xml version="1.0" encoding="UTF-8"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules"
  xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules
      http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules/EventRules.xsd">
     <eventRule name="SCENARIO1_XAL" ruleType="filter" isDraft="yes">
     <eventCondition name="MTEEventRaised1" eventProvider="SapMonitor"
                     eventType="CCMSEventGenerated">
         <filteringPredicate>
              <attributeFilter name="Workstation" operator="eq"
                    <value>SAP_XA</value>
              </attributeFilter>
              <attributeFilter name="InputSAPSystemID" operator="eq"
                    <value>GS0</value>
              </attributeFilter>
              <attributeFilter name="InputMonitoringContextName" operator="eq"
                    <value>ALE/EDI GS0(000) Log.sys TVALE</value>
              </attributeFilter>
              <attributeFilter name="InputMonObjectName" operator="eq"
                    <value>Inbound IDoc ORDER_IDOC</value>
              </attributeFilter>
              <attributeFilter name="InputMonFieldName" operator="eq"
                    <value>Inbound: IDoc generated</value>
              </attributeFilter>
         </filteringPredicate>
     </eventCondition>
     <action actionProvider="TWSAction" actionType="sbj" responseType="onDetection">
              <parameter name="JobUseUniqueAlias">
                    <value>false</value>
              </parameter>
              <parameter name="JobDefinitionWorkstationName">
                    <value>SAP_XA</value>
              </parameter>
              <parameter name="JobAlias">
                    <value>IDOC_%{MTEEventRaised1.AlertUID}</value>
              </parameter>
              <parameter name="JobDefinitionName">
                    <value>PROCESS_ORDER</value>
              </parameter>
     </action>
     <action actionProvider="TWSAction" actionType="sbd" responseType="onDetection">
              <parameter name="JobUseUniqueAlias">
                    <value>false</value>
              </parameter>
              <parameter name="JobWorkstationName">
                    <value>TWS_HOST_FTA</value>
              </parameter>
              <parameter name="JobTask">
                    <value>C:\TWS_home\methods\r3batch -t PI
                           -c %{MTEEventRaised1.Workstation} -- "
                           -t CA -ALSYSID %{MTEEventRaised1.AlertSAPSystemID}
                           -MSEGNAME %{MTEEventRaised1.AlertMTESegment}
                           -ALUNIQNUM %{MTEEventRaised1.AlertUID}
                           -ALINDEX %{MTEEventRaised1.AlertIndex}
                           -ALERTDATE %{MTEEventRaised1.AlertDate}
                           -ALERTTIME %{MTEEventRaised1.AlertTime} "
                    </value>
               </parameter>
              <parameter name="JobPriority">
                    <value>10</value>
              </parameter>
              <parameter name="JobType">
                    <value>Command</value>
              </parameter>
              <parameter name="JobAlias">
                    <value>CONFIRM_%{MTEEventRaised1.AlertUID}</value>
              </parameter>
              <parameter name="JobStreamName">
                    <value>CONFIRM_STREAM</value>
              </parameter>
              <parameter name="JobLogin">
                    <value>twsuser</value>
              </parameter>
     </action>
     </eventRule>
</eventRuleSet>
```

National Language support

Using the local and global configuration files, you can set up r3batch to use different code pages and languages for both its output and the connection with a remote R/3 system.

The National Language support feature allows you to install r3batch on a localized Tivoli Workload Scheduler workstation and use localized characters for Tivoli Workload Scheduler job names, job streams, and R/3 variants.

As explained in "Unicode support" on page 78, this version of Tivoli Workload Scheduler for Applications for SAP features Unicode, which is widely supported by R/3 systems since version 4.7. However, if either the workstation running r3batch or the target R/3 systems do not support Unicode, this section describes how you configure code pages and national languages for r3batch.

Setting National Language support options

The following options control the code page and national language used by Tivoli Workload Scheduler for Applications for SAP, when Unicode support is not used:

TWSXA_CP

The code page used to establish the connection between r3batch and the target SAP R/3 system.

If you are running a non-Unicode version of r3batch, set this option to the code page installed on the SAP R/3 system (for a list of the valid code pages, refer to "SAP R/3 supported code pages" on page 208). The default value is the SAP R/3 code page 1100, similar to the standard ISO8859-1. In all other cases, this option is ignored.

TWSXA_LANG

The language that r3batch uses to log in. It can be one of the following (DE, EN, and JA can be set from the Option Editor. The other languages can be set using any text editor):

- Brazilian Portuguese (pt_BR)

- English (EN, the default value)
- French (FR)
- German (DE)
- Italian (IT)
- Japanese (JA)
- Korean (KO)
- Simplified Chinese (zh_CN)
- Spanish (ES)
- Traditional Chinese (zh_TW)

Note: If you are working with InfoPackages and process chains on operating systems that do not support Unicode, this option must be set.

TWSMETH_CP

The code page that `r3batch` uses for its output. The default is the code page used by the Tivoli Workload Scheduler workstation that hosts `r3batch`.

Ensure that the `TWSMETH_CP` and `TWSMETH_LANG` options are consistent.

TWSMETH_LANG

The catalog language used by `r3batch`. The default is the language used by the Tivoli Workload Scheduler workstation that hosts `r3batch`.

Ensure that the `TWSMETH_CP` and `TWSMETH_LANG` options are consistent.

SAP R/3 supported code pages

To communicate with SAP R/3 systems, Tivoli Workload Scheduler for Applications for SAP uses the following code pages. Use these values to set option `TWSXA_CP`, only when `r3batch` does not support Unicode.

Table 61. SAP R/3 supported code pages

SAP R/3 code pages	Description
1100	8859-1, this is the default value
1103	MS 850
8000	SJIS: Shift JIS
8300	BIG5: Traditional Chinese
8400	GBK: Simplified Chinese

Part 5. z/OS access method

Introducing Tivoli Workload Scheduler for Applications for z/OS

What you need to know and to do before using Tivoli Workload Scheduler for Applications for z/OS.

Using Tivoli Workload Scheduler for Applications for z/OS you can schedule and control z/OS jobs using the job scheduling features of Tivoli Workload Scheduler.

Note: Throughout this publication, the term z/OS is used to refer also to supported versions of OS/390®. For detailed information, see "Software requirements."

Software requirements

In addition to the supported operating systems and prerequisite software listed in "Installing, upgrading, and uninstalling IBM Tivoli Workload Scheduler for Applications" on page 3, you must have the following:

Table 62. Additional z/OS software requirements

Software Requirement	Version
TCP/IP	One of the following: • IBM 3.1, or later • Interlink 3.1, or later • Open Connect 2.3, or later
Job Scheduling Interface	• JES2 or JES3. **Note:** In a z/OS Parallel Sysplex® environment, a JES job must be run on the LPAR where it is submitted. • OPC interface: IBM Tivoli Workload Scheduler for z/OS.
Note: Support to IPv6 is provided only with Tivoli Workload Scheduler, version 8.4 or later, for TCP/IP on IBM 3.1, or later.	

Features

Using Tivoli Workload Scheduler for Applications for z/OS you can:

- Use Tivoli Workload Scheduler to schedule z/OS jobs to run at specific times and in a prescribed order.
- Define dependencies between Tivoli Workload Scheduler jobs running on different systems and operating systems.
- Define dependencies for Tivoli Workload Scheduler jobs based on the completion of z/OS jobs that were not launched by Tivoli Workload Scheduler.
- Define dependencies for Tivoli Workload Scheduler jobs based on the existence of files on a z/OS system.

Roles and responsibilities

In a typical enterprise, different users contribute to the implementation and operation of the product. Table 63 on page 210 describes the roles and responsibilities of all the users in the process model, showing the tasks they perform.

Table 63. Roles and responsibilities in Tivoli Workload Scheduler for Applications for z/OS

User role	User task
z/OS administrator	• “Installing, configuring, and uninstalling the z/OS gateway” • “Downloading z/OS gateway fixes from FTP” on page 217
Tivoli Workload Scheduler for Applications configurator	“Defining the configuration options” on page 220
Tivoli Workload Scheduler for Applications developer	“Defining z/OS jobs in Tivoli Workload Scheduler” on page 222

Installing, configuring, and uninstalling the z/OS gateway

Tivoli Workload Scheduler for Applications for z/OS consists of the z/OS access method that must be located on the Tivoli Workload Scheduler host, and of the gateway software that is located on the z/OS system. To install the Tivoli Workload Scheduler for Applications for z/OS access method, refer to “Installing, upgrading, and uninstalling IBM Tivoli Workload Scheduler for Applications” on page 3.

To install, configure, and uninstall the z/OS gateway, refer to the following sections. If you upgraded Tivoli Workload Scheduler for Applications for z/OS to version 8.5, do not install the z/OS gateway.

- “Installing”
- “Configuring” on page 212
- “Uninstalling” on page 213

Installing

You can install the z/OS gateway module in either of these two ways:

- Unload the files from the IBM Tivoli Workload Scheduler for Applications CD. See “Unloading the files from the CD.”
- Unload the files from a 3480 tape cartridge written in non-IDRC (uncompressed) format. See “Unloading the files from the tape” on page 211.

Unloading the files from the CD:
The z/OS gateway files are stored in the `ZOS` directory of the product CD and are named:

- LOADLIB
- SAMPLES

To unload the files onto your z/OS system:

1. From your TSO session emulator, select the ISPF command shell (TSO command) and use the File Transfer utility (Send to Host) to transfer the LOADLIB library and SAMPLES member from the CD to the z/OS system, setting the transfer for a logical record length of 80 and a fixed record format.
2. Receive the members in output data sets using the `INDSN` option. This unloads 12 load modules into the output library and two samples into the sample library.

For example:

LOADLIB

1. Issue the following command:

```
TSO RECEIVE INDSN('TWS4APPS.LOADLIB.L80')
```

A prompt is displayed similar to the following:

```
INMR901I Dataset TWS84.XAGENT.V8R4M0.FIXPAC04.DRV1511.LOADLIB
from TWSUSR2 on NODENAME NMR906A
Enter restore parameters or 'DELETE' or 'END' +
***
```

2. Reply:

   ```
   da('TWS4APPS.LOADLIB')
   ```

   where "da" means "data set" and the MVS™ data set name in quotes is the name you want for the output loadlib data set.

   Some IEBCOPY messages are displayed as the library is uncompressed.

SAMPLIB

1. Issue the following command:

   ```
   TSO RECEIVE INDSN('TWS4APPS.SAMPLIB.L80')
   ```

   A prompt is displayed similar to the following:

   ```
   INMR901I Dataset TWS84.XAGENT.V8R4M0.FIXPAC04.DRV1511.SAMPLIB
   from TWSUSR2 on NODENAME NMR906A
   Enter restore parameters or 'DELETE' or 'END' +
   ***
   ```

2. Reply:

   ```
   da('TWS4APPS.SAMPLIB')
   ```

   where "da" means "data set" and the MVS data set name in quotes is the name you want for the output samplib data set.

   Some IEBCOPY messages are displayed as the library is uncompressed.

Unloading the files from the tape:
The z/OS gateway files are supplied on a 3480 tape cartridge written in non-IDRC (uncompressed) format.

Modify and submit the JCL below to unload the tape. Customize the job card and modify the following parameters according to your environment standards:

- Enter an appropriate job name.
- Identify a 3480 tape device.

```
//MVSXAUNL JOB (876903,D07),'OPCL3',MSGLEVEL=(1,1),
//        MSGCLASS=A,CLASS=A,NOTIFY=&SYSUID
//**********************************************************************
//*                                                                    *
//* THIS IS THE JOB THAT UNLOADS THE TIVOLI WORKLOAD SCHEDULER FOR     *
//* APPLICATIONS z/OS Access Method Version 8.4 TO CUSTOMIZE           *
//*                                                                    *
//**********************************************************************
//STEP01   EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//INDD     DD DSN=TWSX.V8R4M0.SAMPLES,
//            DISP=(OLD,PASS),UNIT=600,
//            VOL=SER=IBM001,
//            LABEL=(1,SL)
//OUTDD    DD DSN=TWSX.V8R4M0.SAMPLES,
//            DISP=(NEW,CATLG),
//            SPACE=(32760,(2,2,10)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=0),
//            UNIT=3390,VOL=SER=OPC00C
//SYSUT3   DD UNIT=SYSDA,SPACE=(TRK,(20,1,10))
```

```
//SYSUT4   DD UNIT=SYSDA,SPACE=(TRK,(20,1,10))
//SYSIN    DD *
         COPY OUTDD=OUTDD,INDD=((INDD,R))
//STEP02   EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//INDD     DD DSN=TWSX.V8R4M0.SERVICE.APFLIB1,
//            DISP=(OLD,PASS),UNIT=600,
//            VOL=SER=IBM001,
//            LABEL=(2,SL)
//OUTDD    DD DSN=TWSX.V8R4M0.SERVICE.APFLIB1,
//            DISP=(NEW,CATLG),
//            SPACE=(32760,(50,50,50)),
//            DCB=(RECFM=U,BLKSIZE=32760),
//            UNIT=3390,VOL=SER=OPC00C
//SYSUT3   DD UNIT=SYSDA,SPACE=(TRK,(20,1,10))
//SYSUT4   DD UNIT=SYSDA,SPACE=(TRK,(20,1,10))
//SYSIN    DD *
         COPY OUTDD=OUTDD,INDD=((INDD,R))
//*
```

Configuring

To configure the z/OS gateway:

1. Authorize the TWS4APPS.LOADLIB in APF, or copy its contents into another authorized load library. See “Setting APF authorizations on z/OS” on page 213 for details.
2. Add EEWTCP00 to the AUTHCMD list in IKJTSO00 (SYS1.PARMLIB)
3. Set the RACF® permissions so that the user who will use the gateway can issue the PARMLIB command. See “Setting RACF authorizations on z/OS” on page 213 for details.
4. Make the IKJTSO00 change effective by performing an IPL, or by issuing the following command:

   ```
   PARMLIB UPDATE(00)
   ```
5. Create a PARMLIB member with the appropriate startup parameters for EEWSPACE and EEWSERVE.

   Note: This member must have the *PACK OFF* option set in its profile. If *PACK ON* is set, the started task will end with *RC=04*.
6. Copy EEWSPACE and EEWSERVE to the PROCLIB (from SAMPLIB) and edit it (for example, include STEPLIB, and specify the appropriate PARMLIB member name).
7. Verify that the TCP/IP port specified in the PARMLIB member is not in use. To do this, issue the following command and review the output:

   ```
   TSO NETSTAT PORTLIST
   ```

   If the port is in use, choose another port that is not in use and modify the PARMLIB member.
8. Ensure that the IEFU84 exit is enabled by checking the SYS1.PARMLIB member SMFPRMxx, or by issuing the following console command:

   ```
   D SMF,O
   ```

   If the SMFPRMxx member must be changed, to make the changes effective issue the command:

   ```
   SET SMF=xx
   ```
9. Set the RACF permissions for the started tasks EEWSPACE and EEWSERVE. For details, see “Setting RACF authorizations on z/OS” on page 213.
10. Start EEWSPACE.

11. When EEWSPACE is up, start EEWSERVE.

Setting APF authorizations on z/OS:
This section describes how to authorize the load library in APF, by following these steps:

1. Issuing the **SETPROG** command from the console log. For example:

   ```
   SETPROG APF,ADD,DSN=twsx.SERVICE.APFLIB1,
   VOL=xxxxxx
   ```

   where: *xxxxxx* is the volume serial number where the load library is located, or:

   ```
   SETPROG APF,ADD,DSN=twsx.SERVICE.APFLIB1,VOL=SMS
   ```

   which indicates a volume controlled by SMS.
2. Update the PROGxx member of SYS1.PARMLIB, or the authorization will be lost at your next IPL.
3. Add EEWTCP00 to the AUTCHCMD NAMES section of SYS1.PARMLIB(IKJTSOxx).

Setting RACF authorizations on z/OS:
This section describes how to set RACF permissions.

To set the RACF permissions to authorize the PARMLIB command to be used by the user "ibmuser", issue the following commands:

```
rdefine tsoauth parmlib uacc(upd)
permit parmlib class(tsoauth) id(ibmuser) acc(upd)
setropts raclist(tsoauth) refresh
```

To set the RACF permissions for the started tasks EEWSPACE and EEWSERVE, issue the following commands:

```
redefine started EEWSPACE.**
   stdata(user(<user_ID>) group(group_name))
redefine started EEWSERVE.**
   stdata(user(<user_ID>) group(group_name))
setropts raclist(started) refresh
```

Uninstalling

To uninstall the z/OS gateway:

1. Stop the started tasks EEWSPACE and EEWSERVE.
2. Cancel the TWS4APPS.LOADLIB library and the TWS4APPS.SAMPLES member.
3. Remove the started task entries from the RACF database by issuing the following commands:

   ```
   rdelete started EEWSPACE.**
   rdelete started EEWSERVE.**
   setropts raclist(started) refresh
   ```

Additional information

The following topics provide additional information for the z/OS extended agent:

- “Gateway software components” on page 214
- “IEFU84 Exit” on page 214
- “Security” on page 214
- “SYSTSIN variables” on page 215
- “SYSTSIN variables” on page 215
- “z/OS gateway version” on page 217

Gateway software components

The two z/OS program components of the Tivoli Workload Scheduler gateway are:

EEWTCP02
: This program establishes that Tivoli Workload Scheduler is tracking on the z/OS system. The program is started by the EEWSPACE job.

EEWTCP00
: This is the “gateway” program that manages TCP/IP communications between Tivoli Workload Scheduler and the z/OS system. It is started by the EEWSERVE job. EEWTCP00 translates Tivoli Workload Scheduler commands to z/OS equivalents, routes z/OS information back to Tivoli Workload Scheduler, and performs EBCDIC-ASCII data conversions.

Both of the programs run as started tasks, with a TIME=NOLIMIT parameter. EEWTCP02 is always started first, followed by EEWTCP00. If the programs must be stopped for any reason, they should be stopped, not canceled, to ensure that they shut down properly without impacting other programs that use the IEFU84 Exit.

IEFU84 Exit

The extended agent for z/OS tracks job streams using the IEFU84 exit. This exit must be turned on in the SMF parm member in SYS1.PARMLIB. IBM distributes a dummy IEFU84 exit with the operating system that is an IEFBR14 program. The EEWSPACE job dynamically chains to the IEFU84 exit. If the IEFU84 exit is currently being used, EEWSPACE will “front-end” the IEFU84 exit, obtain the information it requires, and then branch to the existing user exit. When EEWSPACE is stopped, it removes itself from the chain and restores the chain to its original status. It is important to note that EEWSPACE has no effect on any existing IEFU84 exits, which continue to run normally.

Security

Security is enforced in several areas, usually, RACF, Top Secret, and ACF2. The EEWSERVE job must have the ability to submit jobs that run under the user IDs that are supplied in the JCL to be submitted. The JCL must not contain passwords. This can be authorized by using SURROGAT class resources in RACF, and the equivalents in ACF2 and Top Secret. PROPCNTL class resources in RACF can be used to prevent submitted jobs from running under the EEWSERVE user ID. ACF2 and Top Secret equivalents can also be used. Resource class JESJOBS in RACF, and ACF2 or Top Secret equivalents, can be used to control which job names and user IDs (with or without passwords) can be submitted by EEWSERVE.

Startup

Follow these steps:

1. Customize and start the EEWSPACE procedure (following the commented instructions it contains) to start the extended agent Gateway Data Space. The job must be a started task and must not be canceled. See “SYSTSIN variables” on page 215 for a description of the parameter settings. EEWSPACE creates the Data Space and installs the IEFU84 exit. To stop the job, use the `STOP EEWSPACE`command from any z/OS console.

   Note:

   a. EEWSPACE must be active before EEWSERVE is started.
   b. To shut down, stop EEWSERVE before stopping EEWSPACE.

2. Customize and start the EEWSERVE procedure by following the commented instructions it contains. For a description of the parameter settings, see “SYSTSIN variables.”
3. To stop the job, use the `STOP EEWSERVE` command from any z/OS console.

SYSTSIN variables

Table 64 lists all the SYSTSIN variables and their description. Modify the settings as required for your site configuration. The default values are shown in parentheses.

Table 64. SYSTSIN variables

Variable	Description
COMPLETIONCODE(LASTSTEP)	Specifies the job completion code of a JES multi-step job. This variable can have one of the following values: **LASTSTEP** The completion code for a JES multi-step job is determined by the last run step in the job. This is the default value. **MAXSTEP** The completion code is determined by the highest completion code of any run step in the job. Any nonzero condition code in the last step (if LASTSTEP was selected) or in any step (if MAXSTEP was selected) causes the job to be considered unsuccessfully completed, unless differently specified using the condcode option in the job definition in Tivoli Workload Scheduler. If a step is flushed, it is not taken into consideration when determining the completion code of the job, unless the flushed step is the last one, in which case the job is always considered as completed unsuccessfully.
DEBUG(NO)	If set to YES, it causes the gateway to output diagnostic messages. Use only in coordination with IBM Software Support.
INTERLINKSUBSYSTEM(ACSS)	The name of the subsystem used by Interlink TCP/IP stack. This variable is ignored if Interlink is not used as TCP/IP stack.
JESCMDCHR($)	The job command recognition character. The default is set to dollar sign ($) for JES2 systems and to asterisk (*) for JES3 systems. Change this variable only if a different command recognition character is being used.
JESINTERFACE(CONSOLE)	CONSOLE is the only supported value for JES and it is the default value. Set to NO if you do not use this interface.

Table 64. SYSTSIN variables (continued)

Variable	**Description**
MAXWAIT(500)	The maximum amount of time, in hundredths of a second, to wait for a response to commands. This variable is ignored if you set JESINTERFACE to NO.
MCSSTORAGE(3)	The amount of storage, in megabytes, used by each extended console used by the gateway. This variable is ignored if you set JESINTERFACE to NO.
OPCINTERFACE(PIF)	PIF is the only supported interface for Tivoli Workload Scheduler for z/OS and is the default value for this variable. Set to NO if you do not use this interface.
OPCMSGCLASS(*)	The message class for the dynamically allocated message logs used by Tivoli Workload Scheduler for z/OS. The asterisk causes the class to be set to the same value as EEWSERVE. Ignored if Tivoli Workload Scheduler for z/OS is not used.
OPCSUBSYSTEM(OPCS)	The subsystem name used for communications with the Tivoli Workload Scheduler for z/OS control region.
PEERADDRESS(0:0:0:0:0:0:0:0)	The default, 0:0:0:0:0:0:0:0, permits access by any host. For better security, enter the IP address of the Workload Scheduler host of the z/OS extended agent. **Note:** 1. Depending on the Internet Protocol you are using, specify an IPv4 or IPv6 address by including also the "." (for IPv4) or ":" (for IPv6) between the 0s. 2. If you are migrating from Tivoli Workload Scheduler for Applications 8.2, 8.3, or 8.4 ensure that you modify this variable to comply with the current format. 3. To connect to an extended agent that supports only Internet Protocol version 4, you must use IPv4.
PORT(5000)	The TCP/IP port number used by Tivoli Workload Scheduler and the gateway for communications. This must be the same as the value entered in the TCP/IP address field of the z/OS extended agent workstation definition.
PUTLINE(YES)	When set to YES, it directs trace information to DDNAME SYSTSPRT.
QLIMIT(2000)	The maximum number of messages to be queued to an extended console.
SUBSYS(UNIS)	The prefix used by the extended agent for z/OS as the first four characters of extended console names. It is also used as the first four characters of internal reader DDNAMES. Change only in coordination with IBM Software Support.

Table 64. SYSTSIN variables (continued)

Variable	Description
SVCDUMP(NO)	When set to YES, abends cause a SVC dump. Use only in coordination with IBM Software Support.
TCPIPSTACK(IBM)	The vendor of TCP/IP stack (IBM, INTERLINK, or OPENCONNECT).
TCPNAME(TCPIP)	The name of the TCP/IP address space when the IBM version of TCP/IP stack is used.
TERMINATOR(X'25')	The transaction termination character. Do not change the default unless asked to do so by IBM Software Support.
WTP(NO)	When set to YES, it directs trace information to SYSLOG as write-to-programmer information. This can be used if SYSTSPRT does not meet your needs.
ZOSV1R2(NO)	Set to YES if you are using the JES3 interface with z/OS V1R2 or later. This parameter is ignored if you are not using the JES3 interface, because: • If you are using the JES2 interface, the level of the operating system does not make any difference to the product

z/OS gateway version

The version of EEWSERVE (and EEWSPACE) appears in the first line of the EEWSERVE (and EEWSPACE) log. For example:

```
EEWTCP10 VERSION RELEASE MODIFICATION LEVEL = VxRyMz
```

where *x* is the version, *y* the release, *z* the modification level.

Downloading z/OS gateway fixes from FTP

This section describes how to obtain the files in a fix or fix pack from the IBM software FTP site.

If a fix or fix pack for Tivoli Workload Scheduler for Applications for z/OS, version 8.4 is issued by IBM, you can receive it by downloading the files from the IBM software FTP site.

Download the z/OS gateway fix pack files as follows:

1. Allocate two data sets with LRECL=80 RECFM=80 to hold the downloaded files. For example:
   - TWS4APPS.LOADLIB.L80 (for the load library)
   - TWS4APPS.SAMPLIB.L80 (for the samples library)

   Use the file characteristics shown in Table 65.

Table 65. File characteristics for obtaining the gateway fix pack files by FTP

Characteristic	Value
Organization	PS
Record format	FB

Table 65. File characteristics for obtaining the gateway fix pack files by FTP (continued)

Characteristic	Value
Record length	80
Block size	27920
1st extent cylinders	1
Secondary cylinders	1

Note: These files are not allocated as type PDS but as regular sequential files.

2. Use FTP to retrieve the LOADLIB and SAMPLES fix pack files from the download site, by logging in as anonymous, with your e-mail address as the password. Issue the following commands:

   ```
   tso ftp ftp.software.ibm.com
   anonymous
   your_e-mail_address
   cd software/tivoli_support/patches/patches_8.4.0/
   cd patch_name
   cd ZOS
   bin
   get loadlib_file_name 'TWS4APPS.LOADLIB.L80' (rep
   get samples_file_name 'TWS4APPS.SAMPLIB.L80' (rep
   quit
   ```

   For example, for Fix Pack 01 the variables in this list of commands would have the following values:

   patch_name
   : `8.4.0-TIV-TWSWSE-FP0001`

   loadlib_file_name
   : `LOADLIB_820WSEFP07`

   samples_file_name
   : `SAMPLES_820WSEFP07`

   Note: The data set names in quotes on the get commands (the MVS file names) must match the files that were allocated in step 1 on page 217.

3. The downloaded files are in an 80-byte packed format. To ensure that the files have been downloaded correctly, browse them. The beginning of the output should be similar to the following:

   ```
   BROWSE TWS4APPS.LOADLIB.L80 Line
   \INMR01. ....&......NODENAME......TWSUSR2......A......A......20
   ....... \INMR02..........IEBCOPY......... ........ ....... .....
   ..........."8. .... ................TWS84..XAGENT..V8R4M0..FIXPAC04..DRV1511..LO
   ```

   If it is not, retry the downloads.

4. Receive the downloaded data sets, as follows:

   LOADLIB

   a. Issue the following command:

      ```
      TSO RECEIVE INDSN('TWS4APPS.LOADLIB.L80')
      ```

      A prompt is displayed similar to the following:

      ```
      INMR901I Dataset TWS84.XAGENT.V8R4M0.FIXPAC04.DRV1511.LOADLIB
      from TWSUSR2 on NODENAME NMR906A
      Enter restore parameters or 'DELETE' or 'END' +
      ***
      ```

b. Reply:

```
da('TWS4APPS.LOADLIB')
```

where da means data set and the MVS data set name in quotes is the name you want for the output loadlib data set.

Some IEBCOPY messages are displayed as the library is uncompressed.

SAMPLIB

a. Issue the following command:

```
TSO RECEIVE INDSN('TWS4APPS.SAMPLIB.L80')
```

A prompt is displayed similar to the following:

```
INMR901I Dataset TWS84.XAGENT.V8R4M0.FIXPAC04.DRV1511.SAMPLIB
from TWSUSR2 on NODENAME NMR906A
Enter restore parameters or 'DELETE' or 'END' +
***
```

b. Reply:

```
da('TWS4APPS.SAMPLIB')
```

where da means data set and the MVS data set name in quotes is the name you want for the output **samplib** data set.

Some IEBCOPY messages are displayed as the library is uncompressed.

After receiving the files, the file characteristics change to those shown in Table 66.

Table 66. File characteristics for the LOADLIB file after receiving it

Characteristic	**Value**
Data Set Name	TWS4APPS.LOADLIB
Organization	PO
Record format	U
Record length	0
Block size	32760
1st extent blocks	10
Secondary blocks	5
Data set name type	PDS

Table 67. File characteristics for the SAMPLIB file after receiving it

Characteristic	**Value**
Data Set Name	TWS4APPS.SAMPLIB
Organization	PO
Record format	FB
Record length	80
Block size	27920
1st extent blocks	4
Secondary blocks	1
Data set name type	PDS

Configure the downloaded fix pack files as described in “Configuring” on page 212.

Locating product support

To find the Tivoli Workload Scheduler for Applications support page, see the appendix on support information in *Tivoli Workload Scheduler: Troubleshooting Guide*.

Use the **Search Support** option to search for items of interest to you. Useful terms to enter for Tivoli Workload Scheduler for Applications for z/OS are: "TWS4APPS", message IDs, ABEND codes, "EEWSERVE", "MVS xagent", "TWS applications z/OS", or the interface types ("TWS for z/OS", JES2, JES3).

Configuring the z/OS access method

Tivoli Workload Scheduler launches jobs on an extended agent or a dynamic agent for z/OS workstation. The supported agent for z/OS is defined in a standard Tivoli Workload Scheduler workstation definition, which gives it a name and identifies the access method to be used. The extended agent or dynamic agent for z/OS workstation is a workstation definition linked to an instance of the z/OS system.

To launch a job on a supported agent for z/OS workstation, Tivoli Workload Scheduler runs the access method, passing it information about the job. The access method communicates with the instance of the z/OS host and monitors the job through completion, writing job progress and status information to the job’s standard list file.

See “Defining supported agent workstations” on page 32 to learn how to define a supported agent workstation in Tivoli Workload Scheduler.

Defining the configuration options

Every installed instance of the z/OS access method must have a matching options file. You can create it during or after the installation of the access method with either the Option Editor (see “Setting options with the Option Editor” on page 26) or any common text editor.

The options file must be located on the Tivoli Workload Scheduler hosting computer for the extended agent or dynamic agent in the `*TWS_home*\methods` directory. If you do not create one, the agent uses by default one of the global options files (either `mvsjes.opts` or `mvsopc.opts`).

Table 68 on page 221 describes the options that you can define for the z/OS access method.

Table 68. Tivoli Workload Scheduler for Applications for z/OS access method options

Options File Entries	**Description**
BLOCKTIME=*min*	(Optional) Defines the amount of time, in minutes, the method waits for a response to a status check before timing out. This value must be less than the value of **CHECKINTERVAL** (described below) and of Tivoli Workload Scheduler's local option **bm check status**. Fractional values are accepted; for example, .5 for 30 seconds, or 1.5 for one minute and 30 seconds. The default is 2. **Note:** Only change this option from the default value if specific network problems cause delays in the transmission of data from the z/OS gateway.
CHECKINTERVAL=*min*	(Optional) Defines the polling rate, in minutes, for checking the status of z/OS jobs that were launched by the method. Fractional values are accepted; for example, .5 for 30 seconds, or 1.5 for one minute and 30 seconds. The default is 2. When checking non-Tivoli Workload Scheduler-launched jobs on z/OS that are used as dependencies, the method uses the Tivoli Workload Scheduler local option **bm check status** instead of **CHECKINTERVAL** to determine its polling rate.
CFUSER=*name*	(Optional [1]) Assigns the user name used by the access method to check file dependencies (CF tasks). It can be the same as **LJUSER**.
GSUSER=*name*	(Optional [1]) Assigns the user name used by the access method to check non-Tivoli Workload Scheduler-launched jobs on z/OS that are used as dependencies (GS tasks). The default is **root**.
LJUSER=*name*	(Optional [1]) Assigns the user name used by the access method to launch jobs (LJ tasks). This must be a valid UNIX or Windows user who submits local jobs and can connect to the Tivoli Workload Scheduler z/OS gateway on the z/OS system. The default is the login parameter in the job definition.
RETRYCOUNT=*count*	(Optional) Defines the number of times a status check is attempted before Tivoli Workload Scheduler writes a timeout message to a job's stdlist file, and marks it in the abend state. For details, see options **CHECKINTERVAL** and **BLOCKTIME**. The default is 10.

Note:

1. For best results, allow *LJUSER*, *CFUSER*, and *GSUSER* to take the default values.

Defining jobs in z/OS

In z/OS, jobs that are scheduled with Tivoli Workload Scheduler are defined as described in the following sections:

- “For JES jobs:” on page 222
- “For Tivoli Workload Scheduler for z/OS jobs:” on page 222

Defining z/OS jobs in Tivoli Workload Scheduler

Tivoli Workload Scheduler job definitions are required for each z/OS job you that want to schedule and launch with Tivoli Workload Scheduler. They are defined in the same ways as other Tivoli Workload Scheduler jobs and include job name, user name, special script name options, and optional recovery options.

Remember that you should include neither special characters, other than dash (-) and underscore (_), nor national characters in the Tivoli Workload Scheduler job name. Such characters are not supported in z/OS, and when the Tivoli Workload Scheduler job is passed to z/OS by the access method, z/OS rejects the name and abends the job if it finds them.

See "Defining jobs for supported agents" on page 36 for reference.

Task definition syntax for z/OS jobs scheduled with Tivoli Workload Scheduler

The following are descriptions of the task definition syntax (command line "scriptname") for z/OS jobs that Tivoli Workload Scheduler schedules and launches from extended agent workstations using the **mvsjes** (JES2/3 jobs) or **mvsopc** (IBM Tivoli Workload Scheduler for z/OS jobs) methods.

You specify these task string parameters in the following places when you define their associated Tivoli Workload Scheduler jobs:

- In the **Task string** field of the Task page of the Properties - Job Definition panel, if you use the Dynamic Workload Console
- As arguments of the `scriptname` keyword in the job definition statement, if you use the Tivoli Workload Scheduler command line.
- As arguments of the `JOBCMD` keyword in the `JOBREC` statement in the `SCRIPTLIB` of Tivoli Workload Scheduler for z/OS, if you are scheduling in an end-to-end environment.

For JES jobs::
The syntax is:

```
dataset [<|= condcode]
```

where:

dataset Specifies the JES job data set or the name of a member of a partitioned data set.

condcode
Specifies the condition code that indicates successful job completion. If preceded by <, the condition code must be less than or equal to this value. If preceded by =, the condition code must be equal to this value. If omitted, " = 0000" is used. Note that there must be a space on both sides of the operator (< or =).

Example:

```
gold.apayable.cntl(apayjob1) = 0004
```

For Tivoli Workload Scheduler for z/OS jobs:: The syntax is:

```
appl  [IA(yymmddhhmm)|IATIME(hhmm)] [...]
     [DEADLINE(yymmddhhmm)|DEADLINETIME(hhmm)]
     [PRIORITY(pri)]
     [CPDEPR(Y|N|P|S)]
```

where:

appl The name of the Tivoli Workload Scheduler for z/OS application to be inserted into the current plan.

IA The input arrival date and time in the form: `yymmddhhmm`.

IATIME
The input arrival time in the form: hhmm.

DEADLINE
The deadline arrival date and time in the form: yymmddhhmm.

DEADLINETIME
The deadline arrival time in the form: hhmm.

PRIORITY
The priority (1-9) at which to run the application.

CPDEPR
The current plan dependency resolution selection.

Y Add all successor and predecessor dependencies.

N Do not add any dependencies, (the default.)

P Add predecessor dependencies.

S Add successor dependencies.

For complete descriptions of the parameters, refer to the Tivoli Workload Scheduler for z/OS documentation.

Example:

```
PREFABJOB44 IA(0202181000) PRIORITY(5) CPDEPR(Y)
```

Task definition syntax for other z/OS jobs

The following are descriptions of the task definition syntax (command line *scriptname*) for z/OS jobs that are to be monitored for completion. The completion of these z/OS jobs can be used as *follows* dependencies for Tivoli Workload Scheduler-launched jobs. The jobs to be monitored can be JES or Tivoli Workload Scheduler for z/OS.

Tivoli Workload Scheduler monitors these jobs until their status changes to *success*.

The details of the outcome of such jobs must be checked in the subsystem where the jobs were launched. Tivoli Workload Scheduler only records whether or not these jobs completed successfully. To find the reason for the failed submission or completion of one of these jobs, or to check for dependency failures, work with the host subsystem operator who can obtain this information from the EEWSERVE log.

The basic syntax is:

```
tws-job follows XAname::"mvs-job"
```

where:

tws-job The name of the Tivoli Workload Scheduler job that depends on the completion of the specified z/OS job.

XAname
The name of the Tivoli Workload Scheduler extended agent workstation

associated with the scheduler of the z/OS job, that is, an extended agent defined with the **mvsjes** or **mvsopc** method. The two colons (::) are a required delimiter.

mvs-job
The identification of the z/OS job. This string must be enclosed in quotation marks. See the syntax descriptions below.

For JES jobs::
The syntax is:

```
"jobname[<|= condcode]"
```

where:

jobname
The name of the job in JES.

condcode
The condition code that indicates successful job completion. If preceded by <, the condition code must be less than or equal to this value. If preceded by =, the condition code must be equal to this value. If omitted, “ = 0000” is used. There must be a space on both sides of the operator (< or =).

Example:

```
job5 follows jesworkstation::"apayable = 0004"
```

For Tivoli Workload Scheduler for z/OS jobs::
The syntax is:

```
"application[IA(yymmddhhmm)|IATIME(hhmm)][...]"
          [JOBNAME(jobname)]
          [OPNO(num)]
```

where:

application
The name of the Tivoli Workload Scheduler for z/OS application (job stream) in the current plan.

IA The input arrival date and time.

IATIME
The input arrival time.

JOBNAME
The z/OS job name.

OPNO
The operation number (1-255). If included, the application is considered completed when it reaches this operation number.

For complete descriptions of the parameters, refer to the Tivoli Workload Scheduler for z/OS documentation. For example:

```
joba follows twsworkstation::"PREFABJOB44 IA(0202181000) JOBNAME(PFJ3)"
```

Operational considerations for the EEWSERVE gateway task

The following considerations apply:

EEWSERVE not running
If the EEWSERVE task on the mainframe is not running, and a Tivoli Workload Scheduler job with no follows dependency is submitted from the

extended agent, the job will show an *error* status and will fail, that is, the job will not run after the EEWSERVE task is started. However, if a Tivoli Workload Scheduler job has a follows dependency for an external (non-Tivoli Workload Scheduler) job which runs under JES or Tivoli Workload Scheduler for Applications for z/OS, the internal check job (CJ) command is reissued after EEWSERVE is started. The extended agent workstation still shows its status as *linked* even if EEWSERVE is not running.

For this reason, if a z/OS automation product such as NetView® is available on the mainframe, write a rule to detect any outages of the EEWSERVE task.

Instance limitations in LPARs

Due to the ENQ/DEQ mechanism in use, only one instance of the EEWTCP02 task (default name EEWSPACE) can be run on a z/OS LPAR. If a second instance is started, it fails with RC=04. So even if you use different started task names and PORT numbers, only one instance of EEWSPACE or EEWSERVE can exist concurrently on a z/OS LPAR.

File (OPENS) dependencies syntax on z/OS files

Tivoli Workload Scheduler jobs and job streams can use z/OS files as dependencies. The basic syntax is:

```
tws-job opens XAname#"hlq1.hlq2.hlq3"
```

where:

tws-job The name of the Tivoli Workload Scheduler job dependent on the specified z/OS file.

XAname

The name of the Tivoli Workload Scheduler extended agent workstation associated with the scheduler of the z/OS job, that is, an extended agent defined with the **mvsjes** or **mvsopc** method.

Note: The # sign is a required delimiter.

hlq1.hlq2.hlq3

The identification of the z/OS file in high level qualifier terms.

Note: This string must be enclosed in quotation marks.

For more information, see “Checking for files on z/OS” on page 228.

Reference information

This chapter describes job states when operating on JES and Tivoli Workload Scheduler for z/OS in the Tivoli Workload Scheduler environment.

Technical overview

The z/OS gateway uses an extended MCS console to communicate with JES. The program issues the MCSOPER macro to activate an extended MCS console. The z/OS gateway can then receive messages and command responses by issuing the MCSOPMSG macro, and can issue commands by issuing the MGCRE macro. All the return codes from the extended MCS macros are handled as described in *IBM z/OS Programming: Authorized Assembler Services Reference, Volume 3, SA22-7611*.

Managing JES jobs

The following sections describe how to manage JES jobs.

Launching JES jobs: To launch and monitor a JES job, Tivoli Workload Scheduler passes the name of the JCL data set or partition data set it wants to run to the z/OS gateway, which, in turn, submits it to JES. Submissions occur using dynamically allocated internal readers in JES. The gateway allocates an internal reader at the start of each task and then submits the job to the reader.

When a job is submitted, the job name and JES job ID are also entered in the Tablespace. When an SMF record containing relevant job scheduling data is passed through the IEFU84 exit, the job and condition code information are made available to Tivoli Workload Scheduler. Because Tivoli Workload Scheduler keeps track of both the job name and the JES job ID, it can check for the specific job it submitted. (Currently, the Gateway uses Type 30 SMF records and also subtypes 1, 4, 5.)

Tivoli Workload Scheduler checks submitted jobs periodically to see if they are active. If a Tivoli Workload Scheduler-submitted job is not active and no information about it is found through the IEFU84 exit, the job is marked as **abend** in Tivoli Workload Scheduler displays. This situation might occur if a job fails for security reasons or JCL syntax problems.

JES job states: Table 69 lists JES job states with respect to Tivoli Workload Scheduler.

Table 69. JES job states with respect to Tivoli Workload Scheduler

Tivoli Workload Scheduler Job State	JES Job State	Comment
intro	Not available	Tivoli Workload Scheduler is starting the method.
wait	Queued	Job is queued.
wait	Not available	If the job remains in this state, it might be due to a security violation in z/OS. Check the job on the z/OS system.
exec	Executing	Job is running.
succ	Completed	Job's condition code meets the completion criteria in the Tivoli Workload Scheduler job definition.
abend	Completed	Job condition code does not meet the completion criteria in the Tivoli Workload Scheduler job definition, or a system or user abend has occurred. System abend codes, in hexadecimal, are prefixed with "S", and user abend codes, in decimal, are prefixed with "U". Both types of code are written to the job stdlist file.
extrn	Not available	Status unknown. Can occur only when checking a job that is used as a dependency.

Monitoring JES jobs: The details of the outcome of JES jobs must be requested from the subsystem where these jobs were launched. From Tivoli Workload Scheduler you should only expect to find out if these jobs completed successfully.

To find the reason for a failed submission or failed completion of one of these jobs or to check for dependency failures, the Tivoli Workload Scheduler operator should work with the host subsystem operator who can obtain this information from the EEWSERVE log.

Checking JES jobs: To check a JES job that was not launched by Tivoli Workload Scheduler, the name of the job is passed by Tivoli Workload Scheduler to the gateway. Because Tivoli Workload Scheduler did not submit the job, the JES job ID is not available. The Gateway enters the name in the Tablespace, and waits for information about the job to appear in SMF records passed through the IEFU84 exit.

The IEFU84 exit cannot handle every job without impacting the performance of the entire system because it is invoked for each job running on the z/OS system.

- If the job is not present in the gateway dataspace, the IEFU84 exit does not perform any action.
- If the job is submitted by Tivoli Workload Scheduler, the gateway inserts the job into the dataspace. In this case the IEFU84 exit monitors the status of the job and of each step contained in the job.
- If the job is not submitted by Tivoli Workload Scheduler, the gateway inserts the job into the dataspace only if the gateway receives a request from Tivoli Workload Scheduler to check its status because the job represents an internetwork dependency.

To have the internetwork dependencies of a z/OS job correctly handled by Tivoli Workload Scheduler on the z/OS system, ensure that there are no occurrences of the z/OS job in any job queue including the output queue. If any occurrences of the z/OS job are present then purge them. The internetwork dependencies of a z/OS job are handled by Tivoli Workload Scheduler in the following ways:

- If there are no occurrences of z/OS jobs in the job queues, the gateway inserts the job into the dataspace the first time it receives the request from Tivoli Workload Scheduler to check the job status. The gateway inserts the job into the dataspace with an unknown job ID ready to be monitored.
- When the z/OS job is submitted, the IEFU84 exit finds the job in the dataspace and updates the corresponding entry with the JES job ID. From now on the z/OS job is monitored using the associated JES job ID. If the job completes successfully, the gateway returns the information to Tivoli Workload Scheduler, and the internetwork dependency is correctly resolved.

Managing Tivoli Workload Scheduler for z/OS jobs

The following sections describe how to manage Tivoli Workload Scheduler for z/OS jobs.

Launching Tivoli Workload Scheduler for z/OS jobs: To launch and monitor a Tivoli Workload Scheduler for z/OS job, Tivoli Workload Scheduler passes the application name, and other optional parameters, it wants to run to the z/OS Gateway. If it exists in the Tivoli Workload Scheduler for z/OS database, the application is inserted into the current plan. The input arrival, deadline arrival, priority, and automatic dependency resolution parameters, if included, override any values specified in Tivoli Workload Scheduler for z/OS.

At a rate defined by the **CheckInterval** value in the method options file, Tivoli Workload Scheduler checks the status of the occurrence (application) in Tivoli Workload Scheduler for z/OS.

Tivoli Workload Scheduler for z/OS operation states: Table 70 lists Tivoli Workload Scheduler for z/OS operation states with respect to Tivoli Workload Scheduler.

Table 70. Tivoli Workload Scheduler for z/OS operation states with respect to Tivoli Workload Scheduler

Tivoli Workload Scheduler Job State	Tivoli Workload Scheduler for z/OS Operation State
wait	pending
wait	undecided
wait	wait
wait	ready
exec	started
succ	complete
abend	interrupted
abend	error
abend	deleted
abend	Not applicable
extrn	Not applicable

Tivoli Workload Scheduler for z/OS occurrence states: Table 71 lists Tivoli Workload Scheduler for z/OS operation occurrence states with respect to Tivoli Workload Scheduler.

Table 71. Tivoli Workload Scheduler for z/OS operation occurrence states with respect to Tivoli Workload Scheduler

Tivoli Workload Scheduler Job Stream State	Tivoli Workload Scheduler for z/OS Occurrence State
wait	pending
wait	undecided
exec	started
succ	complete
abend	error
abend	deleted
abend	Not applicable
extrn	Not applicable. Status unknown. Can occur only when checking a job that is used as a dependency.

Checking Tivoli Workload Scheduler for z/OS jobs: To check a Tivoli Workload Scheduler for z/OS job that was not launched by Tivoli Workload Scheduler, the name of the application, and optionally the operation, is passed to the gateway. A check is made to see if the occurrence or operation is in the current plan. If it is not found, Tivoli Workload Scheduler rechecks at a rate defined by the **bm check status** value in its local options file.

Checking for files on z/OS

The existence of a file can be used as a job dependency in Tivoli Workload Scheduler. To check for the existence of a file on z/OS, Tivoli Workload Scheduler passes the file name to the Gateway. The file name is allocated with DISP=OLD, and is considered to exist if the following conditions apply:

- The data is cataloged
- It is allocated
- It is not being used by another task

If the data set does not exist, Tivoli Workload Scheduler continues to wait and check for the file at a frequency determined by the `bm check file` option in the `localopts` file of the fault-tolerant workstation that is hosting the extended agent. The localopts options are described in the *Tivoli Workload Scheduler: Planning and Installation Guide*.

Note: Tivoli Workload Scheduler can only use fully qualified data set names for non-partitioned files. If a Generation Data Group name is to be used, it must be the fully qualified name and not a relative name (for example, xxxxx.xxxxx(-1) cannot be used).

Timing considerations

When Tivoli Workload Scheduler checks dependencies on z/OS jobs *not* launched by Tivoli Workload Scheduler for Applications, certain timing issues are critical to ensuring that any associated job dependencies are correctly resolved. For the correct resolution of these external dependencies, Tivoli Workload Scheduler must attempt to resolve the dependency at least once *before* the z/OS job is submitted. After the z/OS job has been submitted and has successfully completed, the next periodic check of the dependency by Tivoli Workload Scheduler can manage the dependency.

External dependencies are checked by Tivoli Workload Scheduler in the following circumstances:

- If the external dependency on a Tivoli Workload Scheduler for Applications z/OS job is the only dependency in the job, the dependency is checked immediately after the plan is generated or extended. However, as the plan might have a period of several days, weeks, or months, this might not be very frequently.
- Subsequently to the first check, the dependency is checked with a frequency determined by the Tivoli Workload Scheduler configuration parameters.
- If the external dependency is preceded by another dependency, the external dependency is not checked until the other dependency is resolved.

Thus, when creating external dependencies in Tivoli Workload Scheduler on Tivoli Workload Scheduler for Applications z/OS jobs, you should schedule the job so that the first dependency check takes place before the z/OS job is submitted.

If this synchronization is not taken into account, Tivoli Workload Scheduler might wait indefinitely to resolve a job dependency. A similar problem can occur as the result of a communication failure between the z/OS and Tivoli Workload Scheduler environments that prevents Tivoli Workload Scheduler from determining the status of a z/OS job to satisfy a job dependency.

Diagnostic information

z/OS jobs submitted by Tivoli Workload Scheduler can fail to complete for a number of reasons. The step in the submission process in which a job fails determines how much information is available and is provided by Tivoli Workload Scheduler as follows:

- If a job fails before it is actually initiated (usually the result of a JCL or security problem), Tivoli Workload Scheduler recognizes that it no longer exists, and marks it as abend in the conman command line displays. No further information is provided.
- If a job fails after being started, Tivoli Workload Scheduler:
  1. Obtains its condition code and user abend code, if any
  2. Writes them to the job standard list file
  3. Marks the job as **abend** in the conman command line or Dynamic Workload Console displays

  Job standard lists can be displayed with the conman command line and with the Dynamic Workload Console.

Troubleshooting

To assist in troubleshooting, ensure that the JES log is obtained for the EEWSPACE and EEWSERVE started tasks. This helps in determining the context in which a message was issued. Depending on the job scheduling interface you use, additional helpful information might be obtained from other logs.

```
EEWI27I APPLICATION application WAS INSERTED IN CP WITH INPUT
ARRIVAL DATE AND TIME yymmddhhss

EEWI28W yymmdd hhmmss APPLICATION appl WAS NOT INSERTED IN
 CURRENT PLAN WITH INPUT ARRIVAL DATE AND TIME iadatetime

EEWI29I yymmdd hhmmss TASK task MODULE module ISSUED, MACRO macro NEAR LABEL label
 WITH RETURN CODE = code AND ERROR NUMBER = err

EEWI30S yymmdd hhmmss module CA7SPAN MUST BE 4 DIGITS IN FORMAT HHMM

EEWI31E TASK task MODULE module LAUNCH OF JOB 'jobname' FAILED

EEWI32S yymmdd hhmmss module AT LEAST ONE INTERFACE MUST BE DIFFERENT FROM NO

EEWI33W yymmdd hhmmss TASK task APPLICATION application NOT FOUND

EEWI34W APPLICATION application NOT FOUND

EEWI35W JCL dataset(member) NOT FOUND

EEWI36W yymmdd hhmmss IA and IATIME cannot be specified together

EEWI37W yymmdd hhmmss DEADLINE and DEADLINETIME cannot be specified together

EEWI38I jobname(jobid) nl n2 result (restype)
```

- n1 indicates the number of seconds passed from the request
- n2 indicates the number of seconds of CPU time consumed
- *result* can assume one of the following values:

  ABEND
  : If the job abends. In this case *restype* can be:

    Sxyz In case of System Abend

    Unnnn
    : in case of User Abend

  CONDCOD
  : If the job does not end successfully due to the condition code of one step that does not match the definition. In this case *restype* contains: the RC=nnnnn value of the return code of the last step that has been run, if LASTSTEP specified, or of the worst step that does not match the definition of the job on the distributed side.

  EXEC If the job is running or is in the input queue.

  JCLERRO
  : If the job failed due to a JCL error.

SUCCES
If the job completed successfully.

UNKNOWN
If the jobid is unknown

blank In case of an internetwork dependency when the manual submission was not performed.

Note: All the above messages are written in the EEWSPACE or in the EEWSERVE log files. These are the files indicated in the SYSTSPRT DD card of the respective procedure. In the files the messages are written starting from column 1, except for the messages that do not contain the date and time after the message identifier, for example EEWI27I. These messages appear with different characteristics in the z/OS system and on the Tivoli Workload Scheduler. In the z/OS system log the messages appear in the text of another message and in some cases they might appear truncated. This is because the maximum length of each text record is limited to 251 characters. In Tivoli Workload Scheduler they are always displayed in their complete form.

The module names are:

Module Name	Description
EEWPRE84	Previous SMF exit
EEWTCP00	Main processing routine
EEWTCP01	Server test vehicle
EEWTCP02	Main dataspace construction routine
EEWTCP05	Establish ESTAE routine to issue failure message
EEWTCP07	Find data set name associated with open ACB/DCB
EEWTCP08	Find data set name associated with DDNAME
EEWTCP10	Initialize and examine variables
EEWTCP15	Locate and initialize control blocks
EEWTCP20	Locate and initialize control blocks
EEWTCP21	Client test vehicle
EEWTCP22	Client test vehicle
EEWTCP23	Client test vehicle
EEWTCP25	Concurrent server
EEWTCP26	Concurrent server Interlink TCP/IP provider
EEWTCP27	Concurrent server for Openconnect TCP/IP stack
EEWTCP30	Child server task IBM Tivoli Workload Scheduler
EEWTCP31	Child server task for Interlink TCP/IP stack
EEWTCP32	Child server task
EEWTCP84	SMF record exit for SMFEWTM
EEWTCP87	IBM Tivoli Workload Scheduler interface
EEWTCP88	IBM Tivoli Workload Scheduler interface post
EEWTCP89	CA-7 BTI
EEWTCP90	Issue command and wait for response
EEWTCP91	Issues TERROR to format messages and issues WTOS

Module Name	Description
EEWTCP92	Find substring within string
EEWTCP93	Issue CA-7 command and wait for response
EEWTCP94	STIMER exit to post ECB past in parm
EEWTCP95	Calls IKJEFF02 to extract buffer using EEWTCP99
EEWTCP96	ESTAE routine to issue failure message
EEWTCP97	Calls IKJEFF19 (DAIRFAIL/VSAMFAIL) to issue message
EEWTCP98	Calls IKJEFF02 to issue message from EEWTCP99
EEWTCP99	Message table
EEWTCPC7	Check CA-7 job
EEWTCPCA	Check application request handler
EEWTCPCF	Check file request handler
EEWTCPCJ	Check job request handler
EEWTCPCO	Check operation request handler
EEWTCPCV	Enumerate job
EEWTCPED	String encrypt/decrypt
EEWTCPEJ	Enumerate job request handler
EEWTCPFJ	Free job request handler
EEWTCPIA	Insert application
EEWTCPIE	String initialization
EEWTCPKJ	Kill job request handler
EEWTCPKO	Delete IBM Tivoli Workload Scheduler application
EEWTCPL7	Launch CA-7 job request handler
EEWTCPLJ	Launch job request handler
EEWTCPLO	Insert IBM Tivoli Workload Scheduler application
EEWTCPQ7	Query CA-7 job request handler
EEWTCPQJ	Query job request handler
EEWTCPQO	Query IBM Tivoli Workload Scheduler application
EEWTCPQS	Query subsystem request handler
EEWTCPRC	Return code
EEWTCPSC	System abend
EEWTCPSE	String encryption/decryption
EEWTCPUC	User abend
EEWTCPWT	Wait for specified amount of time
EEWTCPXX	Invalid request handler

APAR PQ88345 add note below.

To assist in troubleshooting, be sure to obtain the JES log for the EEWSPACE and EEWSERVE started tasks. This will help in determining the context in which a message was issued. Depending on the job scheduling interface you use, additional helpful information may be obtained from other logs. For example, if you use CA-7, you should obtain the following:

- The CA-7 log
- The console log for the interval covering the test period
- The job log of the job resulting in error (if this is the case)
- The UNIX script file related to that job

Part 6. Managing InfoSphere DataStage jobs

Overview

By using the IBM Tivoli Workload Scheduler Plug-in for InfoSphere DataStage you schedule IBM InfoSphere DataStage jobs with Tivoli Workload Scheduler. In this way, you take advantage of all the Tivoli Workload Scheduler functions to manage InfoSphere DataStage jobs.

You can manage these jobs both in a distributed and in a z/OS environment, by selecting the appropriate engine.

Business scenario

A retail company has many shops around the world. Each shop has its own local database which stores daily transactions and tracks the number of articles left in its store. The central business division of the company needs to analyze every morning all the reports that show the number of articles sold in every country, grouped by predefined categories.

The company collects this data by using InfoSphere DataStage. The company runs InfoSphere DataStage jobs overnight to collect data from the local database of each store and then, using the procedures stored in the central database, produces the aggregated data to create the business reports. The process that runs the InfoSphere DataStage jobs overnight is performed manually by an operator. To reduce costs and to ensure that the SLA requirement of having data available every morning is satisfied, the company wants to automate this process.

Using Tivoli Workload Scheduler Plug-in for InfoSphere DataStage, the company can satisfy this objective because the product helps it to automate and control the entire process.

Software requirements

You use Tivoli Workload Scheduler Plug-in for InfoSphere DataStage both in a distributed and in a z/OS environment. You must install the Tivoli Workload Automation product appropriate for your environment, and the InfoSphere DataStage Server product. To install InfoSphere DataStage Plug-in, see “Installing and uninstalling IBM Tivoli Workload Scheduler for Applications Plug-ins” on page 41. In details:

Distributed **To manage InfoSphere DataStage jobs, install:**

- The Tivoli Workload Scheduler master domain manager
- A dynamic agent connected:
  - To the master domain manager

  Or

  - To a dynamic domain manager connected to the master domain manager.
- Tivoli Workload Scheduler for Applications on the same workstation where you installed:
  - The master domain manager
  - The dynamic domain manager

  - The dynamic agent
- InfoSphere DataStage Server version 8.5 on the same workstation where you installed the dynamic domain manager or the dynamic agent.

z/OS **To manage InfoSphere DataStage jobs, install:**

- The Tivoli Workload Scheduler for z/OS controller.
- A Tivoli Workload Scheduler for z/OS agent connected:
  - To the Tivoli Workload Scheduler for z/OS controller.

  Or

  - To a dynamic domain manager connected to the Tivoli Workload Scheduler for z/OS controller.
- Tivoli Workload Scheduler for Applications on the same workstation where you installed:
  - The Tivoli Workload Scheduler for z/OS connector.
  - The dynamic domain manager.
  - Tivoli Workload Scheduler for z/OS agent.
- InfoSphere DataStage Server version 8.5 on the same workstation where you installed the Tivoli Workload Scheduler for z/OS agent.

For detailed information about the Tivoli Workload Scheduler supported operating systems, see the System Requirements Document. For detailed information about the InfoSphere DataStage supported operating systems and installation requirements, see the InfoSphere DataStage Information Center.

Tasks

Using Tivoli Workload Scheduler Plug-in for InfoSphere DataStage, you can perform on InfoSphere DataStage jobs all the tasks you can perform on Tivoli Workload Scheduler jobs by using the Dynamic Workload Console, the command line, or the ISPF interface.

Roles and responsibilities

In a typical enterprise, different users contribute to the implementation and operation of the product. Table 72 describes the roles and responsibilities of all actors in the process model, showing the tasks they perform.

Table 72. Roles and responsibilities in Tivoli Workload Scheduler for InfoSphere DataStage

User role	User task
InfoSphere DataStage administrator	Performing the installation of the InfoSphere DataStage Server.
InfoSphere DataStage developer	Creating the InfoSphere DataStage jobs.
InfoSphere DataStage configurator	"Customizing Tivoli Workload Scheduler to run InfoSphere DataStage jobs" on page 237
Tivoli Workload Scheduler IT administrator	Performing the installation of the Tivoli Workload Scheduler environment.
Tivoli Workload Scheduler job scheduler	"Defining a Tivoli Workload Scheduler job to run an InfoSphere DataStage job" on page 238
Tivoli Workload Scheduler scheduling operator	"Monitoring InfoSphere DataStage jobs" on page 244

Customizing Tivoli Workload Scheduler to run InfoSphere DataStage jobs

You can customize Tivoli Workload Scheduler for running InfoSphere DataStage jobs by using the DataStageJobExecutor.properties file.

The DataStageJobExecutor.properties file is a text file that contains the user credentials and the path where you installed the InfoSphere DataStage Server.

The file is located in the directory:

```
agent_install_dir/TWA/TWS/JavaExt/cfg
```

Where *agent_install_dir* is the path where you installed the Tivoli Workload Scheduler dynamic agent or the Tivoli Workload Scheduler for z/OS agent.

You can define the property contained in the .properties file, except the **installDir** property, at job definition time also. In this case Tivoli Workload Scheduler uses the values you specify at job definition time for running the job. You must define all or none of these properties either using the command line, the Dynamic Workload Console, or the .properties file, otherwise you receive an error message. If you do not define any of these properties either in the .properties file or at job definition time, Tivoli Workload Scheduler assumes that you did not set any security on the InfoSphere DataStage server and tries the connection to the InfoSphere DataStage server anyway. Table 73 describes the properties contained in DataStageJobExecutor.properties.

Table 73. Properties to run InfoSphere DataStage jobs

Property	Description/value	Required
installDir	The InfoSphere DataStage Server directory where you find the dsjob command. The default is: **UNIX and Linux operating systems:** **/opt/IBM/InformationServer/Server/DSEngine/bin** **Windows operating systems:** **C:/IBM/InformationServer/Server/DSEngine/bin**	✓
Domain	The domain to log on to.	
Server	The server to log on to.	
UserName	The user to use when logging on.	
Password	The password of the authorized user. It is encrypted when you retrieve any information by using a list or when you submit the job.	

Example

This example shows a .properties file.

```
installDir=C:/IBM/InformationServer/Server/DSEngine/bin
Domain=nc112206.rmlb.it.ibm.com:9444
Server=nc112206
UserName=isadmin
Password=ispass
```

Defining a Tivoli Workload Scheduler job to run an InfoSphere DataStage job

Define Tivoli Workload Scheduler jobs to run InfoSphere DataStage jobs by using either the Dynamic Workload Console or the command line.

Distributed Define a Tivoli Workload Scheduler job to run an InfoSphere DataStage job by using the Dynamic Workload Console connected to a distributed engine or by using the **composer** command line.

z/OS Define a Tivoli Workload Scheduler job to run an InfoSphere DataStage job by using the Dynamic Workload Console connected to a z/OS engine.

Defining a Tivoli Workload Scheduler job to run an InfoSphere DataStage job by using composer

Distributed

Define a Tivoli Workload Scheduler job to run an InfoSphere DataStage job by using **composer** if you are connected to a distributed engine.

Purpose

Use the syntax described below to define a Tivoli Workload Scheduler job to run an InfoSphere DataStage job.

Syntax

```
$jobs
[workstation#]jobname
     task job_definition [streamlogon username]}
   [description "description"]
   [tasktype tasktype]
   [interactive]
   [rccondsucc "Success Condition"]
   [recovery
         {stop | continue | rerun}
         [after [workstation#]jobname]
         [abendprompt "text"] ]
```

Use the **task** argument, specifying the XML syntax for the InfoSphere DataStage.

For a detailed description of the XML syntax, see "Job definition for InfoSphere DataStage jobs."

For the job definition, see Defining objects in the database> Defining Scheduling Object > Job Definition in the *Tivoli Workload Scheduler User's Guide and Reference*.

Job definition for InfoSphere DataStage jobs

Tivoli Workload Scheduler job definition properties and JSDL examples for running InfoSphere DataStage jobs.

Table 74 on page 239 describes the required and optional attributes for InfoSphere DataStage jobs, together with a description of each attribute.

Table 74. Required and optional attributes for the job definition of InfoSphere DataStage jobs.

Attribute	Description/value	Required
application name	datastage	✓
Domain	The domain to log on to. See **Note**.	
Server	The server to log on to. See **Note**.	
UserName	The user to use when logging on. See **Note**.	
password	The password of the authorized user. It is encrypted when you submit the job. See **Note**.	
ProjectName	The name of the project containing the job.	✓
JobName	The name of the job to run.	Required if you do not specify the job alias
JobAlias	The alias associated to the job to run.	Required if you do not specify the job name
FileRemotePath	The fully qualified path to the file that contains the parameter values to pass to the job.	
ParameterTableValues	The list of parameters to associate to the job.	
ForceReset	Specify it to reset the InfoSphere DataStage job before it runs. When an InfoSphere DataStage job has a status of **Crashed** or **Aborted**, you must reset it before running the job again.	
WarningLimit	Specify the maximum number of warnings that the InfoSphere DataStage job can reach before failing. Specify a value from 1 through 9999999. The default value is **50**.	
RowLimit	Specify the maximum number of rows to process in an InfoSphere DataStage job before ending. Possible values are from 1 through 9999999. The default value is **1000**.	

Table 74. Required and optional attributes for the job definition of InfoSphere DataStage jobs. (continued)

Attribute	Description/value	Required
DisableProjectErrorMessageHandler	Specify it to disable any error message handler that was set on a project. A message handler defines rules about how to handle messages generated when a parallel job is running. You can, for example, use a message handler to specify that certain types of message must not be written to the log. You define a project level message handler in the InfoSphere DataStage Administrator, and it applies to all parallel jobs within the specified project.	
DisableJobErrorMessageHandler	Specify it to disable any error message handler that was set on a job. From the InfoSphere DataStage Designer, you can specify that any existing handler applies to a specific job. When you compile the job, the handler is included in the job executable as a local handler. In this case, it can be exported to other systems if necessary.	
UseDefault	Specify it to use the default values set for the project to generate the operational metadata when the InfoSphere DataStage job runs. It is the default.	
GenerateOperationalMetadata	Specify it to generate operational metadata when running the InfoSphere DataStage job. Operational metadata describe the events and processes that occur and the objects that are affected when you run the InfoSphere DataStage job.	
DoNotGenerateOperationalMetadata	Specify it to not generate operational metadata when running the InfoSphere DataStage job.	

Note: If you do not want to specify this attribute in the XML, you can define it in the `DataStageJobExecutor.properties` file. You must define all or none of these values otherwise you receive an error message. See “Customizing Tivoli Workload Scheduler to run InfoSphere DataStage jobs” on page 237.

The following example shows the job definition of an InfoSphere DataStage job with only the required attributes specified:

```
NC112206#DS01
 TASK
   <?xml version="1.0" encoding="UTF-8"?>
```

```
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
   xmlns:jsdldatastage="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdldatastage"
   name="DATASTAGE">
  <jsdl:application name="datastage">
    <jsdldatastage:datastage>
      <jsdldatastage:DataStageParameters>
         <jsdldatastage:DataStagePanel>
              <jsdldatastage:Logon>
                 <jsdldatastage:Domain>it112206.rome.it.com:9444</jsdldatastage:Domain>
                 <jsdldatastage:Server>it112206</jsdldatastage:Server>
                 <jsdldatastage:UserName>userName</jsdldatastage:UserName>
                 <jsdldatastage:password>password</jsdldatastage:password>
              </jsdldatastage:Logon>
              <jsdldatastage:JobDefinitionGroup>
                <jsdldatastage:ProjectNameGroup>
                  <jsdldatastage:ProjectName>DatastageReport</jsdldatastage:ProjectName>
                </jsdldatastage:ProjectNameGroup>
                    <jsdldatastage:JobNameButtonGroup>
                          <jsdldatastage:JobNameRadioButton>
                              <jsdldatastage:JobName>dsj01_succ</jsdldatastage:JobName>
                          </jsdldatastage:JobNameRadioButton>
                    </jsdldatastage:JobNameButtonGroup>
                      <jsdldatastage:FileRemotePath/>
              </jsdldatastage:JobDefinitionGroup>
              <jsdldatastage:JobExecutionGroup/>
         </jsdldatastage:DataStagePanel>
         <jsdldatastage:OptionsPanel>
                 <jsdldatastage:JobOptionsGroup>
                         <jsdldatastage:WarningLimitButtonGroup>
                                 <jsdldatastage:NoWarningLimitButton/>
                         </jsdldatastage:WarningLimitButtonGroup>
                         <jsdldatastage:RowLimitButtonGroup>
                                 <jsdldatastage:NoRowLimitButton/>
                         </jsdldatastage:RowLimitButtonGroup>
                         <jsdldatastage:OperationalMetadataGroup>
                                 <jsdldatastage:UseDefault/>
                         </jsdldatastage:OperationalMetadataGroup>
                 </jsdldatastage:JobOptionsGroup>
         </jsdldatastage:OptionsPanel>
      </jsdldatastage:DataStageParameters>
    </jsdldatastage:datastage>
  </jsdl:application>
</jsdl:jobDefinition>
 RECOVERY STOP
```

The following example shows the job definition of an InfoSphere DataStage job with all the attributes specified:

```
NC112206#DS01
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
 xmlns:jsdldatastage="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdldatastage" name="DATASTAGE">
  <jsdl:application name="datastage">
   <jsdldatastage:datastage>
       <jsdldatastage:DataStageParameters>
            <jsdldatastage:DataStagePanel>
                <jsdldatastage:Logon>
                      <jsdldatastage:Domain>it112206.rome.it.com:9444</jsdldatastage:Domain>
                      <jsdldatastage:Server>it112206</jsdldatastage:Server>
                      <jsdldatastage:UserName>userName</jsdldatastage:UserName>
                      <jsdldatastage:password>password</jsdldatastage:password>
                 </jsdldatastage:Logon>
                 <jsdldatastage:JobDefinitionGroup>
                      <jsdldatastage:ProjectNameGroup>
                              <jsdldatastage:ProjectName>DatastageReport</jsdldatastage:ProjectName>
                      </jsdldatastage:ProjectNameGroup>
                      <jsdldatastage:JobNameButtonGroup>
                                 <jsdldatastage:JobAliasRadioButton>
                                      <jsdldatastage:JobAlias>dsj01_succ_id</jsdldatastage:JobAlias>
                                 </jsdldatastage:JobAliasRadioButton>
                      </jsdldatastage:JobNameButtonGroup>
                         <jsdldatastage:FileRemotePath>/opt/remoteFile</jsdldatastage:FileRemotePath>
                         <jsdldatastage:ParameterTableValues>
                            <jsdldatastage:ParameterTableValue key="waitTime">10
                               </jsdldatastage:ParameterTableValue>
                            <jsdldatastage:ParameterTableValue key="date">2011-09-14
                               </jsdldatastage:ParameterTableValue>
                         </jsdldatastage:ParameterTableValues>
                      </jsdldatastage:JobDefinitionGroup>
                      <jsdldatastage:JobExecutionGroup>
                              <jsdldatastage:ForceReset/>
                      </jsdldatastage:JobExecutionGroup>
              </jsdldatastage:DataStagePanel>
              <jsdldatastage:OptionsPanel>
                     <jsdldatastage:JobOptionsGroup>
                              <jsdldatastage:WarningLimitButtonGroup>
                                      <jsdldatastage:WarningLimitButton>
                                  <jsdldatastage:WarningLimit>40</jsdldatastage:WarningLimit>
                                      </jsdldatastage:WarningLimitButton>
                              </jsdldatastage:WarningLimitButtonGroup>
                               <jsdldatastage:RowLimitButtonGroup>
                                      <jsdldatastage:RowLimitButton>
                                              <jsdldatastage:RowLimit>250</jsdldatastage:RowLimit>
                                      </jsdldatastage:RowLimitButton>
                               </jsdldatastage:RowLimitButtonGroup>
                              <jsdldatastage:DisableProjectErrorMessageHandler/>
                              <jsdldatastage:DisableJobErrorMessageHandler/>
                               <jsdldatastage:OperationalMetadataGroup>
                                       <jsdldatastage:GenerateOperationalMetadata/>
                               </jsdldatastage:OperationalMetadataGroup>
                      </jsdldatastage:JobOptionsGroup>
            </jsdldatastage:OptionsPanel>
       </jsdldatastage:DataStageParameters>
 </jsdldatastage:datastage>
</jsdl:application>
</jsdl:jobDefinition>
 RECOVERY STOP
```

Defining Tivoli Workload Scheduler jobs to run InfoSphere DataStage jobs by using the Dynamic Workload Console

You define jobs by using the Dynamic Workload Console when you are working in either a distributed environment or in a z/OS environment.

To define a job that runs an InfoSphere DataStage job by using the Dynamic Workload Console, perform the following procedure.

1. In the console navigation tree, expand **Workload** > **Design** and click **Create Workload Definitions**
2. Specify an engine name, either distributed or z/OS. The Workload Designer is displayed.
3. In the Working List panel, select **New** > **Job Definition** > **Business Analytics** > **DataStage**. The properties of the job are displayed in the right-hand panel for editing.
4. In the properties panel, specify the attributes for the job definition you are creating. You can find detailed information about all the attributes in the help available with the panel. In particular:

   In the General panel:

   Distributed **Environment:**
   Enter the name of the Tivoli Workload Scheduler job that runs the InfoSphere DataStage job.

   Enter the name of the workstation where you installed the Tivoli Workload Scheduler agent and the InfoSphere DataStage server.

   z/OS **Environment:**
   Enter the name of the partitioned data set where you want to create the JCL.

   Enter the name of the JCL you want to create in the partitioned data set.

   Enter the name of the workstation where you installed the Tivoli Workload Scheduler agent and the InfoSphere DataStage server.

   In the DataStage panel:

In the Credentials section:

Enter the credentials related to the InfoSphere DataStage job. If you do not want to specify them here, you can define them in the `DataStageJobExecutor.properties` file. In this case Tivoli Workload Scheduler reads them from the `.properties` file when you retrieve any information by using a list or when you submit the job.

If you do not specify them either using the Dynamic Workload Console or in the `.properties` file, Tivoli Workload Scheduler assumes that you did not set any security on the InfoSphere DataStage server and tries the connection to the InfoSphere DataStage server anyway. You must specify all or none of these values either using the Dynamic Workload Console or the `.properties` file, otherwise you receive an error message. See "Customizing Tivoli Workload Scheduler to run InfoSphere DataStage jobs" on page 237.

In the Job Definition section

Enter the project name and the job name or select them from the appropriate lists. Tivoli Workload Scheduler retrieves this information directly from the InfoSphere DataStage server database. Alternatively you can use the job alias.

You can view the list of parameters defined for the InfoSphere DataStage job. Select the ones you want to define for the job and associate a value to them.

Select **Reset job before running** to reset the InfoSphere DataStage job before it runs. When an InfoSphere DataStage job has a status of **Crashed** or **Aborted**, you must reset it before running the job again.

In the Options panel:

Use this panel to define the options for the InfoSphere DataStage job.

5. Click **Save** to save the job definition in the database.

Scheduling and submitting job streams for InfoSphere DataStage jobs

Submitting Tivoli Workload Scheduler for InfoSphere DataStage jobs.

Distributed You use the Dynamic Workload Console or the command line.

z/OS You use the Dynamic Workload Console or the ISPF application.

After you define a Tivoli Workload Scheduler for InfoSphere DataStage job, you add it to a job stream with all the necessary scheduling arguments and submit it. After submission you can kill the Tivoli Workload Scheduler for InfoSphere DataStage job if necessary, this action is converted in a **Stop** action for the InfoSphere DataStage job.

If the Tivoli Workload Scheduler agent becomes not available when you submit the Tivoli Workload Scheduler for InfoSphere DataStage job or when the job is running, Tivoli Workload Scheduler collects the job log when the agent restarts and assigns the **Error** or **ABEND** status to the Tivoli Workload Scheduler for InfoSphere DataStage job whatever is the status of the InfoSphere DataStage job.

Monitoring InfoSphere DataStage jobs

Monitor jobs by using the Dynamic Workload Console, the command line, or the ISPF application.

Distributed You monitor jobs by using the Dynamic Workload Console connected to a distributed engine or by using the **conman** command line.

z/OS You monitor jobs by using the Dynamic Workload Console connected to a z/OS engine or the ISPF application.

For detailed information about:

How to monitor jobs by using the Dynamic Workload Console
See the online help or Monitoring your Objects in the Plan > Monitoring your Workload >Creating a task to monitor jobs in the *Dynamic Workload Console User's Guide*.

How to monitor jobs by using conman
See Managing objects in the plan - conman in *Tivoli Workload Scheduler User's Guide and Reference*.

How to monitor jobs by using the ISPF application
See Monitoring the Workload in *Tivoli Workload Scheduler for z/OS Managing the Workload*.

Mapping between Tivoli Workload Scheduler and InfoSphere DataStage job statuses

Map Tivoli Workload Scheduler job status to InfoSphere DataStage job status to understand their processing.

Table 75 table shows how you can map the Tivoli Workload Scheduler job status to the InfoSphere DataStage job status based on the return code you find in the job log output.

Table 75. Mapping between Tivoli Workload Scheduler and InfoSphere DataStage job statuses

InfoSphere DataStage job return code	InfoSphere DataStage command line job status	InfoSphere DataStage director job status	Dynamic Workload Console job status	Tivoli Workload Scheduler job status	Tivoli Workload Scheduler for z/OS job status
0	RUNNING	Running	Running	EXEC	Executing
1	RUN OK	Finished	Successful	SUCC	Completed
2	RUN with WARNINGS	Finished (see log)	Successful	SUCC	Completed
3	RUN FAILED	Abort Error	Error	ABEND	Error
96	UNKNOWN	Crashed	Error	ABEND	Error
97	STOPPED	Stopped	Error	ABEND	Error
98	NOT RUNNABLE	Not compiled	Error	FAILED	Error

Any other return code or status you find in the InfoSphere DataStage log generated either by using the InfoSphere DataStage command line or the InfoSphere DataStage Directory interface, is mapped to **error** if you are using the Dynamic Workload Console or to **FAILED** if you are using the Tivoli Workload Scheduler command line.

Analyzing the job log

When a job runs Tivoli Workload Scheduler creates a job log that you can analyze to verify the job status.

Distributed You analyze the job log by using the Dynamic Workload Console or the **conman** command line.

z/OS You analyze the job log by using the Dynamic Workload Console or the ISPF application.

For detailed information about:

How to analyze the job log using the Dynamic Workload Console
See the online help or **Monitoring your Objects in the Plan** > **Monitoring your Workload** > **Creating a task to monitor jobs** in the *Dynamic Workload Console User's Guide*.

How to analyze the job log using conman
See **Managing objects in the plan - conman** > **Conman commands** > **showjobs** in the *Tivoli Workload Scheduler User's Guide and Reference*.

How to analyze the job log using the ISPF application
See Monitoring the Workload in the *Tivoli Workload Scheduler for z/OS Managing the Workload* manual.

Job log output

The Tivoli Workload Scheduler for InfoSphere DataStage job log and its content.

Purpose

The output of a Tivoli Workload Scheduler for InfoSphere DataStage job is composed of two parts:

- The first part is the result of the InfoSphere DataStage **dsjob -logsum** command.
- The second part is the result of the InfoSphere DataStage **dsjob -report DETAIL** command.

The output shows the current job status.

Sample

This example shows the output of a job that completed successfully:

```
================================================================
0       RESET   Wed Oct 05 17:15:44 2011
        Log cleared by user
1       STARTED Wed Oct 05 17:18:28 2011
        Starting Job dsj01_succ.
2       INFO    Wed Oct 05 17:18:28 2011
        Environment variable settings:  (...)
3       INFO    Wed Oct 05 17:18:28 2011
        dsj01_succ: Set NLS locale to US-ENGLISH,US-ENGLISH,US-ENGLISH,
        US-ENGLISH,US-ENGLISH
4       INFO    Wed Oct 05 17:18:31 2011
        dsj01_succ..DB2_UDB_API_2.IDENT2:
        DSD.StageRun Active stage starting, tracemode = 0.
5       INFO    Wed Oct 05 17:18:31 2011
        dsj01_succ..DB2_UDB_API_2.IDENT2: NLS stage locale is
          US-ENGLISH,US-ENGLISH,US-ENGLISH,US-ENGLISH,US-ENGLISH
6       INFO    Wed Oct 05 17:18:31 2011
        dsj01_succ..InputDataFile: Using project default NLS map MS1252
```

```
7       INFO    Wed Oct 05 17:18:31 2011
        dsj01_succ..DB2_UDB_API_2: Using NLS map MS1252
8       INFO    Wed Oct 05 17:18:35 2011
        dsj01_succ..DB2_UDB_API_2: Executing BeforeSQL
9       INFO    Wed Oct 05 17:18:37 2011
        dsj01_succ..DB2_UDB_API_2:
        Transaction committed as part of link close processing.
10      INFO    Wed Oct 05 17:18:37 2011
        dsj01_succ..DB2_UDB_API_2.IDENT2: Stage statistics  (...)
11      INFO    Wed Oct 05 17:18:37 2011
        dsj01_succ..DB2_UDB_API_2.IDENT2: DSD.StageRun Active stage finishing.
12      INFO    Wed Oct 05 17:18:38 2011
        dsj01_succ..OutputDataFile.IDENT1:
        DSD.StageRun Active stage starting, tracemode = 0.
13      INFO    Wed Oct 05 17:18:38 2011
        dsj01_succ..OutputDataFile.IDENT1: NLS stage locale is
         US-ENGLISH,US-ENGLISH,US-ENGLISH,US-ENGLISH,US-ENGLISH
14      INFO    Wed Oct 05 17:18:38 2011
        dsj01_succ..DB2_UDB_API_2: Using NLS map MS1252
15      INFO    Wed Oct 05 17:18:40 2011
        dsj01_succ..OutputDataFile: Using project default NLS map MS1252
16      WARNING Wed Oct 05 17:18:41 2011
        Link report for link dsj01_succ.DB2_UDB_API_2.DSLink3 not found.
17      INFO    Wed Oct 05 17:18:41 2011
        dsj01_succ..OutputDataFile.IDENT1: Stage statistics  (...)
18      INFO    Wed Oct 05 17:18:41 2011
        dsj01_succ..OutputDataFile.IDENT1: DSD.StageRun Active stage finishing.
19      STARTED Wed Oct 05 17:18:42 2011
        Finished Job dsj01_succ.
**************************************************
STATUS REPORT FOR JOB: dsj01_succ
Generated: 2011-10-05 17:19:11
   Job start time=2011-10-05 17:18:28
   Job end time=2011-10-05 17:18:41
   Job elapsed time=00:00:13
   Job status=2 (Finished with warnings)
      Stage: DB2_UDB_API_2.IDENT2, 10 rows input
      Stage start time=2011-10-05 17:18:37, end time=2011-10-05 17:18:37,
      elapsed=00:00:00
         Link: DSLink7, 10 rows
         Link: DSLink7, 10 rows
      Stage: OutputDataFile.IDENT1, 10 rows input,
           last error=2011-10-05 17:18:41\NC112206|isadmin\2\Link
           report for link dsj01_succ.DB2_UDB_API_2.DSLink3 not found.
      Stage start time=2011-10-05 17:18:41, end time=2011-10-05 17:18:41,
      elapsed=00:00:00
         Link: DSLink3, 10 rows,
         last error=2011-10-05 17:18:41\NC112206|isadmin\2\Link
         report for link dsj01_succ.DB2_UDB_API_2.DSLink3 not found.
         Link: DSLink3, 10 rows,
         last error=2011-10-05 17:18:41\NC112206|isadmin\2\Link
         report for link dsj01_succ.DB2_UDB_API_2.DSLink3 not found.
```

Analyzing the InfoSphere DataStage job properties

When the job is running you can analyze its properties to verify its status.

The job properties output is the result of the InfoSphere DataStage **dsjob -jobinfo** command. This example shows the properties of a job that completed successfully with warnings.

```
Job properties
dsjob.exe" -domain nc112206.romelab.it.ibm.com:9444
 -server nc112206 -user isadmin -password isadmin -jobinfo tws4apps dsj01_succ
...
Extra Information
```

```
Job Status      : RUN with WARNINGS (2)
Job Controller  : not available
Job Start Time  : Wed Oct 05 17:18:28 2011
Job Wave Number : 142
User Status     : not available
Job Control     : 0
Interim Status  : NOT RUNNING (99)
```

Part 7. Managing Cognos reports

Overview

Use the IBM Tivoli Workload Scheduler Application Plug-in for Cognos to run IBM Cognos version 10.1 reports, interactive reports, query, and report views, using Tivoli Workload Scheduler. In this way, you take advantage of all the Tivoli Workload Scheduler scheduling functions to manage these reports.

You manage these reports both in a distributed and in a z/OS environment, by selecting the appropriate engine.

From now on, any reference in the documentation to reports is intended to mean reports, interactive reports, query, and report views, unless otherwise specified.

Business scenario

A retail company has many shops around the world. Each shop has its own local database, which stores daily transactions and tracks the number of articles remaining in stock. The central business division of the company every morning needs to analyze all the reports that show the number of articles sold in every country, grouped by predefined categories.

The company collects this data by using InfoSphere DataStage and creates the reports using Cognos. Overnight, the company runs the following processes:

- InfoSphere DataStage jobs to collect data from the local database of each store and then, using the procedures stored in the central database, to produce the aggregated data to create the business reports.
- Cognos jobs to create the business reports to be used by the business analysts.

Both processes are performed manually by an operator. To reduce costs and to ensure that the SLA requirement of having data available every morning is satisfied, the company wants to automate the entire process.

Using Tivoli Workload Scheduler Plug-in for InfoSphere DataStage and for Cognos, the company can satisfy this objective because the product provides the plug-in necessary to automate and control the entire process.

Software requirements

You use Tivoli Workload Scheduler Plug-in for Cognos both in a distributed and in a z/OS environment. You must install the Tivoli Workload Automation product appropriate for your environment, and the Cognos Server product. To install Cognos Plug-in, see “Installing and uninstalling IBM Tivoli Workload Scheduler for Applications Plug-ins” on page 41. In details:

Distributed **To manage Cognos reports, install:**

- A master domain manager.
- A dynamic agent connected to:
  - The master domain manager.

  Or

  - A dynamic domain manager connected to the master domain manager.
- Tivoli Workload Scheduler for Applications on the same workstation where you installed:
  - The master domain manager.
  - The dynamic domain manager.
  - The dynamic agent.
- Cognos Server version 10.1.

z/OS **To manage Cognos reports, install:**

- The controller.
- The Tivoli Workload Scheduler for z/OS connector.
- A Tivoli Workload Scheduler for z/OS agent connected:
  - To the controller

  Or

  - To a dynamic domain manager connected to the controller
- Tivoli Workload Scheduler for Applications on the same workstation where you installed:
  - The Tivoli Workload Scheduler for z/OS connector
  - The dynamic domain manager
  - TheTivoli Workload Scheduler for z/OS agent
- Cognos Server version 10.1.

For detailed information about the Tivoli Workload Scheduler supported operating systems, see the System Requirements Document. For detailed information about the Cognos supported operating systems and installation requirements, see the Cognos Information Center.

Tasks

Using Tivoli Workload Scheduler Plug-in for Cognos, you can perform on Cognos reports all the tasks you can perform on Tivoli Workload Scheduler jobs by using the Dynamic Workload Console, the command line, or the ISPF interface.

Roles and responsibilities

In a typical enterprise, different users contribute to the implementation and operation of the product. Table 76 describes the roles and responsibilities of all actors in the process model, showing the tasks they perform.

Table 76. Roles and responsibilities in Tivoli Workload Scheduler for Cognos

User role	User task
Cognos administrator	Performing the installation of the Cognos Server.
Cognos developer	Creating the Cognos reports.
Cognos configurator	“Customizing Tivoli Workload Scheduler to run Cognos reports” on page 251
Tivoli Workload Scheduler IT administrator	Performing the installation of the Tivoli Workload Scheduler environment.
Tivoli Workload Scheduler job scheduler	“Defining a Tivoli Workload Scheduler job to run a Cognos report” on page 254

Table 76. Roles and responsibilities in Tivoli Workload Scheduler for Cognos (continued)

User role	User task
Tivoli Workload Scheduler scheduling operator	"Monitoring Tivoli Workload Scheduler jobs that run Cognos reports" on page 262

Customizing Tivoli Workload Scheduler to run Cognos reports

You can customize Tivoli Workload Scheduler to run Cognos report by using the `CognosJobExecutor.properties` file.

The `CognosJobExecutor.properties` file is a text file that contains the server credentials, the authentication information, and the dispatch servlet name you specified when configuring the Cognos product.

The file is located in the directory:

```
agent_install_dir/TWA/TWS/JavaExt/cfg
```

Where *agent_install_dir* is the path where you installed the Tivoli Workload Scheduler dynamic agent or the Tivoli Workload Scheduler for z/OS agent.

You can define the properties contained in the `.properties` file, except the **dispatchServlet** value, at job definition time also. In this case Tivoli Workload Scheduler uses the values you specify at job definition time for running the report. Table 77 describes the properties contained in `CognosJobExecutor.properties`.

Table 77. Properties for running Cognos reports

Property	Description/value	Required
namespace	The Cognos namespace to log on to. You do not need to specify this value if you set the **true** value for the **Allow anonymous access?** property in the **IBM Cognos Configuration** > **Local Configuration** > **Security** > **Authentication** > **Cognos** window.	
username	The user to use when logging on to the namespace. You do not need to specify this value if you set the **true** value for the **Allow anonymous access?** property in the **IBM Cognos Configuration** > **Local Configuration** > **Security** > **Authentication** > **Cognos** window.	
password	The password of the authorized user. It is encrypted in the agent configuration file when you retrieve any information by using a list, or when you submit the report. You do not need to specify this value if you set the **true** value for the **Allow anonymous access?** property in the **IBM Cognos Configuration** > **Local Configuration** > **Security** > **Authentication** > **Cognos** window.	
serverAddress	The host name or TCP/IP address of the Cognos server you want to connect to.	

Table 77. Properties for running Cognos reports (continued)

Property	Description/value	Required
serverPort	The port to use for the connection. The default value is **9300**. If you use SSL, the default is **9334**.	
dispatchServlet	The dispatch you specified in the **IBM Cognos Configuration** > **Environment** > **Dispatcher Settings** window. The default value is **/p2pd/servlet/dispatch**.	✓

Example

This example shows a `CognosJobExecutor.properties` file.

```
namespace=cognosNamespace
username=userName
password=userPwd
serverAddress=nc112006
serverPort=9300
dispatchServlet=/p2pd/servlet/dispatch
```

Configuring the agent to use Cognos in SSL

Configure the Tivoli Workload Scheduler agent to connect to a Cognos server that is using SSL.

To configure the agent to connect to a Cognos server that is using SSL, perform the following procedure.

Note: On Windows operating systems, path with blanks must be specified between double quotes.

1. On the Cognos server, run the following command to export the certificate:

   On Windows operating systems:

   ```
   <Cognos_inst_path>\bin\ThirdPartyCertificateTool.bat -E -T
    -r \<certificate_dir>\<certificate_name>
    -k <Cognos_inst_path>\configuration\signkeypair\jCAKeystore
    -p <cognos_keystore_password>
   ```

   On UNIX and Linux operating systems:

   ```
   <Cognos_inst_path>/bin/ThirdPartyCertificateTool -E -T
    -r /<certificate_dir>/<certificate_name>
    -k <Cognos_inst_path>/configuration/signkeypair/jCAKeystore
    -p <cognos_keystore_password>
   ```

   where:

   Cognos_inst_path
   Specify the path where you installed the Cognos server.

   certificate_dir
   Specify the directory in which to export the Cognos certificate.

   certificate_name
   Specify the name of the Cognos certificate you export.

   cognos_keystore_password
   Specify the Cognos password defined in the **IBM Cognos Configuration** > **Security** > **Cryptography** > **Cognos - Certificate Authority settings - Certificate Authority key store password**.

For example, if you installed the Cognos server on a UNIX operating system in the `/opt/ibm/Cognos/c10` path, you want to export the `/tmp/cacert.cer` certificate and the Certificate Authority key store password is **pass00w0rd**, run the command as follows:

```
/opt/ibm/cognos/c10/bin/ThirdPartyCertificateTool.sh -E -T
 -r /tmp/cacert.cer
 -k /opt/ibm/cognos/c10/configuration/signkeypair/jCAKeystore
 -p pass00w0rd
```

2. On the agent, run the following command to import the certificate into the agent keystore:

On Windows operating systems:

```
<agent_inst_path>\TWS\JavaExt\jre\jre\bin\keytool -import
 -file <exp_certificate_dir>\<certificate_name>
 -keystore <agent_inst_path>\TWS\JavaExt\jre\jre\lib\security\cacerts
 -storepass <keystore_password> -alias Cognos10
```

On UNIX and Linux operating systems:

```
<agent_inst_path>/TWS/JavaExt/jre/jre/bin/keytool -import
 -file <exp_certificate_dir>/<certificate_name>
 -keystore <agent_inst_path>/TWS/JavaExt/jre/jre/lib/security/cacerts
 -storepass <keystore_password> -alias Cognos10
```

where:

agent_inst_path
Specify the path where you installed the agent.

exp_certificate_dir
Specify the directory where you stored the exported Cognos certificate.

certificate_name
Specify the certificate you want to import.

keystore_password
Specify the keystore password of the Java extension. The default is **changeit**.

For example, if you installed the agent on a Windows operating system in the `D:\TWS\Engine\tws_user\` path, you want to import the **cacert.cer** certificate in the `c:\app\certificate\` directory, and the password of the agent keystore is **a0password**, run the command as follows:

```
D:\TWS\Engine\tws_user\TWS\JavaExt\jre\jre\bin\keytool
 -import -file c:\app\certificate\cacert.cer
 -keystore D:\TWS\Engine\tws_user\TWS\JavaExt\jre\jre\lib\security\cacerts
 -storepass a0password -alias Cognos10
```

3. In the agent `JobManager.ini` configuration file, add the **JVMOptions** parameter as follows:

```
JVMOptions  = -Djavax.net.ssl.trustStore=
"<agent_inst_path>\TWS\JavaExt\jre\jre\lib\security\cacerts"
 -Djavax.net.ssl.trustStorePassword=<keystore_password>
```

On Windows operating systems:

```
JVMOptions  = -Djavax.net.ssl.trustStore=
"<agent_inst_path>\TWS\JavaExt\jre\jre\lib\security\cacerts"
 -Djavax.net.ssl.trustStorePassword=<keystore_password>
```

On UNIX and Linux operating systems:

```
JVMOptions  = -Djavax.net.ssl.trustStore=
"<agent_inst_path>/TWS/JavaExt/jre/jre/lib/security/cacerts"
 -Djavax.net.ssl.trustStorePassword=<keystore_password>
```

where:

agent_inst_path
: Specify the path where you installed the agent.

keystore_password
: Specify the keystore password of the Java extension.

For example, if you installed the agent on a Windows operating system in the `D:\TWS\Engine\tws_user\` path, the agent keystore path is `D:\TWS\Engine\tws_user\TWS\JavaExt\jre\jre\lib\security\cacerts` and the password agent keystore is **a0password**, add the **JVMOptions** parameter as follows:

```
JVMOptions  = -Djavax.net.ssl.trustStore=
"D:\TWS\Engine\tws_user\TWS\JavaExt\jre\jre\lib\security\cacerts"
 -Djavax.net.ssl.trustStorePassword=a0password
```

Defining a Tivoli Workload Scheduler job to run a Cognos report

Define Tivoli Workload Scheduler jobs to run Cognos reports by using either the Dynamic Workload Console or the command line.

Distributed Define a Tivoli Workload Scheduler job to run a Cognos report by using the Dynamic Workload Console connected to a distributed engine or the **composer** command line.

z/OS Define a Tivoli Workload Scheduler job to run a Cognos report by using the Dynamic Workload Console connected to a z/OS engine.

Defining jobs by using composer

Distributed

Define a Tivoli Workload Scheduler job to run a Cognos report by using **composer** if you are connected to a distributed engine.

Purpose

Use the following syntax to define a Tivoli Workload Scheduler job to run a Cognos report.

Syntax

```
$jobs
[workstation#]jobname
    task job_definition [streamlogon username]}
   [description "description"]
   [tasktype tasktype]
   [interactive]
   [rccondsucc "Success Condition"]
   [recovery
        {stop | continue | rerun}
        [after [workstation#]jobname]
        [abendprompt "text"] ]
```

Use the **task** argument, specifying the XML syntax for the Cognos report.

For a detailed description of the XML syntax, see “Job definition for Cognos reports” on page 255.

For the job definition, see Defining objects in the database> Defining Scheduling Object > Job Definition in the *Tivoli Workload Scheduler User's Guide and Reference*.

Job definition for Cognos reports

Job definition properties and JSDL examples for Cognos reports.

Table 78 describes the required and optional attributes for the definition of jobs to run Cognos reports, together with a description of each attribute.

Table 78. Required and optional attributes for the definition of jobs running Cognos reports.

Attribute	Description/value	Required
Namespace	The Cognos namespace to log on to. It contains the Cognos objects, such as groups, roles, data sources, distribution lists, and contacts. You do not have to specify this value if: • You specified it in the `CognosJobExecutor.properties` file. • You set the **true** value for the **Allow anonymous access?** property in the **IBM Cognos Configuration** > **Local Configuration** > **Security** > **Authentication** > **Cognos** window.	
UserName	The user to use when logging on to the namespace. You do not have to specify this value if: • You specified it in the `CognosJobExecutor.properties` file. • You set the **true** value for the **Allow anonymous access?** property in the **IBM Cognos Configuration** > **Local Configuration** > **Security** > **Authentication** > **Cognos** window.	
password	The password of the authorized user. It is encrypted when you submit the job. You do not have to specify this value if: • You specified it in the `CognosJobExecutor.properties` file. • You set the **true** value for the **Allow anonymous access?** property in the **IBM Cognos Configuration** > **Local Configuration** > **Security** > **Authentication** > **Cognos** window.	
serverAddress	The host name or TCP/IP address of the Cognos server you want to connect to. You do not have to specify this value if: • You specified it in the `CognosJobExecutor.properties` file. • You set the **true** value for the **Allow anonymous access?** property in the **IBM Cognos Configuration** > **Local Configuration** > **Security** > **Authentication** > **Cognos** window.	

Table 78. Required and optional attributes for the definition of jobs running Cognos reports. (continued)

Attribute	Description/value	Required
port	The port to use for the connection. The default value is **9300**. If you select Use SSL, the default is **9334**. For detailed information, see the Cognos documentation. You do not have to specify this value if: • You specified it in the `CognosJobExecutor.properties` file. • You set the **true** value for the **Allow anonymous access?** property in the **IBM Cognos Configuration** > **Local Configuration** > **Security** > **Authentication** > **Cognos** window.	
SslCheck	Specify it to use SSL connection security to communicate with the Cognos server. If you do not specify it the communication is HTTP.	
reportPath	The path to the report that you want to run.	✓
parametersValues	The list of parameters to associate to the job. When you specify parameters that can have multiple values, use **-Value:** as separator. For example, you can write as follows: `Rome-Value:Paris` Parameters of type date, time, and time stamp, must be specified using a specific syntax. See "Specifying the parameter format for date, time, and time stamp" on page 260. Values for parameters that require parameterized filters must be specified using the procedure described in "Specifying parameterized filters" on page 261.	
outputFile	The fully qualified path where you want to save the report. The path must exist and you must have the permission to write in it. The report is saved with the extension corresponding to the format you specified. Tivoli Workload Scheduler saves, in the path you specify, the report with its content. You must specify at least the path or an email address for the delivery options otherwise you receive an error message when you save the job definition.	

Table 78. Required and optional attributes for the definition of jobs running Cognos reports. (continued)

Attribute	Description/value	Required
formatTypes	The format in which you want to generate the report. For reports, supported formats are: • CSV • Excel 2002 • Excel 2007 • HTML • HTML Fragment • HTML Web Archive • PDF • XML For interactive reports and views of interactive reports you can use only the HTML format. Even if, for a view of an interactive report, you select a format different from HTML, the product saves the output in HTML format when the job runs. The default value is **HTML**.	
emailGroup	The delivery options for the Cognos report. If you did not insert a value for the **Output file** attribute, you must enter at least an email address. The email delivery status does not impact the job completion status. **To, Cc, Bcc** The email addresses to which you want to send the report. Address the message by typing at least an email address in one of the address fields, To, Cc (carbon copy), or Bcc (blind carbon copy). To specify multiple addresses divide them by using commas. **Subject** The subject of the email. **Body** The body of the email.	

The following example shows the job definition for a Cognos report with only the required attributes specified:

```
$JOBS
NC125152#COGNOS_ONLY_REQUIRED_FIELDS
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
 xmlns:jsdlcognos="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdlcognos" name="COGNOS">
  <jsdl:application name="cognos">
    <jsdlcognos:cognos>
   <jsdlcognos:CognosParameters>
    <jsdlcognos:CognosPanel>
     <jsdlcognos:credentialsGroup>
      <jsdlcognos:namespace/>
      <jsdlcognos:userName/>
      <jsdlcognos:password/>
     </jsdlcognos:credentialsGroup>
     <jsdlcognos:serverConnectionGroup>
          <jsdlcognos:serverAddress/>
          <jsdlcognos:port/>
          <jsdlcognos:CheckSSLGroup/>
     </jsdlcognos:serverConnectionGroup>
          <jsdlcognos:reportGroup>
           <jsdlcognos:ReportPathGroup>
            <jsdlcognos:reportPath>
                        Address Report - Path:/content/package[@name='tws4apps']
                        /report[@name='Address Report']
                    </jsdlcognos:reportPath>
           </jsdlcognos:ReportPathGroup>
          <jsdlcognos:outputFile>C:\outputFile</jsdlcognos:outputFile>
           <jsdlcognos:formatTypes>PDF</jsdlcognos:formatTypes>
     </jsdlcognos:reportGroup>
    </jsdlcognos:CognosPanel>
    <jsdlcognos:OptionPanel>
     <jsdlcognos:emailGroup>
          <jsdlcognos:toAddress/>
          <jsdlcognos:ccAddress/>
          <jsdlcognos:bccAddress/>
          <jsdlcognos:subject/>
         <jsdlcognos:body/>
     </jsdlcognos:emailGroup>
    </jsdlcognos:OptionPanel>
   </jsdlcognos:CognosParameters>
  </jsdlcognos:cognos>
  </jsdl:application>
</jsdl:jobDefinition>
 RECOVERY STOP
```

The following example shows the job definition for a Cognos report with all the attributes specified:

```
$JOBS
NC125152#REPFOREUROPEBUSINESS COGNOS_ALL_FIELDS
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
 xmlns:jsdlcognos="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdlcognos" name="COGNOS">
  <jsdl:application name="cognos">
    <jsdlcognos:cognos>
   <jsdlcognos:CognosParameters>
    <jsdlcognos:CognosPanel>
     <jsdlcognos:credentialsGroup>
      <jsdlcognos:namespace>NTLM</jsdlcognos:namespace>
      <jsdlcognos:userName>Administrator</jsdlcognos:userName>
      <jsdlcognos:password>{aes}SgB6gmS+3xj0Yq2QsINVOtsNCeZIIsMwt08kwO6ZCR4=
              </jsdlcognos:password>
     </jsdlcognos:credentialsGroup>
     <jsdlcognos:serverConnectionGroup>
      <jsdlcognos:serverAddress>nc112006</jsdlcognos:serverAddress>
      <jsdlcognos:port>9300</jsdlcognos:port>
      <jsdlcognos:CheckSSLGroup>
       <jsdlcognos:SslCheck/>
      </jsdlcognos:CheckSSLGroup>
     </jsdlcognos:serverConnectionGroup>
     <jsdlcognos:reportGroup>
      <jsdlcognos:ReportPathGroup>
       <jsdlcognos:reportPath>date and time report - in values -
                    Path:/content/package[@name='cognosTime']/interactiveReport
                    [@name='date and time report - in values']
                </jsdlcognos:reportPath>
      </jsdlcognos:ReportPathGroup>
      <jsdlcognos:parametersValues>
       <jsdlcognos:parametersValue key="Date prompt">Between 2012-02-03 and 2012-04-15
                </jsdlcognos:parametersValue>
       <jsdlcognos:parametersValue key="Time prompt">-Value:01:00:00-Value:01:01:01
                </jsdlcognos:parametersValue>
       <jsdlcognos:parametersValue key="Date and time prompt">
                 2012-02-03 15:05:00-Value:2012-02-03T16:01:00-Value:2012-02-03T16:00:00
                </jsdlcognos:parametersValue>
      </jsdlcognos:parametersValues>
      <jsdlcognos:outputFile>C:\Cognos reports\ReportForEuropeBusiness
             </jsdlcognos:outputFile>
      <jsdlcognos:formatTypes>HTML</jsdlcognos:formatTypes>
     </jsdlcognos:reportGroup>
    </jsdlcognos:CognosPanel>
    <jsdlcognos:OptionPanel>
     <jsdlcognos:emailGroup>
      <jsdlcognos:toAddress>gsg@it.im.com,igg@it.im.com</jsdlcognos:toAddress>
      <jsdlcognos:ccAddress>ibf@it.ibm.com</jsdlcognos:ccAddress>
      <jsdlcognos:bccAddress>igg@it.ibm.com</jsdlcognos:bccAddress>
      <jsdlcognos:subject>Report from Cognos</jsdlcognos:subject>
      <jsdlcognos:body>Report generated by Cognos:</jsdlcognos:body>
     </jsdlcognos:emailGroup>
    </jsdlcognos:OptionPanel>
   </jsdlcognos:CognosParameters>
  </jsdlcognos:cognos>
  </jsdl:application>
</jsdl:jobDefinition>
 RECOVERY STOP
```

Defining jobs by using the Dynamic Workload Console

You define Tivoli Workload Scheduler jobs to run Cognos reports by using the Dynamic Workload Console when you are working in either a distributed or in a z/OS environment.

To define a job by using the Dynamic Workload Console, perform the following procedure.

1. In the console navigation tree, expand **Workload** > **Design** and click **Create Workload Definitions**.
2. Specify an engine name, either distributed or z/OS. The Workload Designer opens.
3. In the Working List panel, select:

   z/OS

   New > **Business Analytics** > **Cognos**

   Distributed

   New > **Job Definition** > **Business Analytics** > **Cognos**

   The properties of the job are displayed in the right-hand panel for editing.
4. In the properties panel, specify the attributes for the job definition you are creating. You can find detailed information about all the attributes in the help available with the panel. In particular:

   In the General panel:

   Distributed **Environment:**

   Enter the name of the Tivoli Workload Scheduler job that runs the Cognos report.

   Enter the name of the workstation where you installed the Tivoli Workload Scheduler agent.

   z/OS **Environment:**

   Enter the name of the partitioned data set where you want to create the JCL.

   Enter the name of the JCL you want to create in the partitioned data set.

   Enter the name of the workstation where you installed the Tivoli Workload Scheduler agent.

   In the Cognos panel:

   In the Credentials section:

   Enter the namespace and the credentials related to the Cognos report. If you do not want to specify them here, you can define them in the `CognosJobExecutor.properties` file. In this case Tivoli Workload Scheduler reads them from the `.properties` file when you retrieve any information by using a list or when you submit the job.

You do not need to specify this value if you set the **true** value for the **Allow anonymous access?** property in the **IBM Cognos Configuration** > **Local Configuration** > **Security** > **Authentication** > **Cognos** window.

In the Server connection section:

Enter the host name or TCP/IP address and the port of the Cognos server you want to connect to. If you do not want to specify them here, you can define them in the `CognosJobExecutor.properties` file.

Select **Use SSL** connection security to communicate in SSL with the Cognos server. If you do not select it the communication is HTTP.

In the Report section:

Enter the path to the report you want to run, the fully qualified path where you want to save the report, the format in which you want to generate the report and parameters if required for the report.

The path must exist and you must have the permission to write in it otherwise you receive an error message.

When you specify parameters that can have multiple values, use **-Value:** as separator. For example, you can write as follows:

```
Rome-Value:Paris
```

Parameters of type date, time, and time stamp, must be specified using a specific syntax. See “Specifying the parameter format for date, time, and time stamp.”

Values for parameters that require parameterized filters must be specified using the procedure described in “Specifying parameterized filters” on page 261.

In the Delivery options panel:

Use this panel to define the options to specify the delivery options for the Cognos report. If you did not insert a value in the **Output file** field you must enter at least an email address.

5. Click **Save** to save the job definition in the database.

Specifying the parameter format for date, time, and time stamp

When defining reports either using **composer** or the Dynamic Workload Console, specify parameters of type date, time, and time stamp, using a specific syntax.

Table 79 on page 261 table shows the syntax you must use when defining reports containing date, time, and time stamp formats as parameters.

Table 79. Examples to use for parameters of date, time, and time stamp formats

		Cognos parameter format examples		
Prompt type	**Cognos parameter format**	**Single value**	**List of values**	**Interval values**
Date	CCYY-MM-DD	2012-02-03	2012-02-03-Value:2012-03-14	Between 2012-02-03 and 2012-04-15
Time	hh:mm:ss	01:00:00	01:00:00-Value:01:01:01	Between 01:00:00 and 23:59:30

Time Stamp	`CCYY-MM-DDThh:mm:ss`
	`or`
	`CCYY-MM-DD hh:mm:ss`
		`2012-02-03 15:05:00`	2012-02-03 15:05:00-Value:2012-02-03T16:01:00-Value:2012-02-03T16:00:00	`Between 2012-02-03 15:05:00 and 2012-04-15T16:00:00`

Note: You must specify the parameter format exactly as they are shown in the table respecting lower case and upper case formats.

Specifying parameterized filters

When defining reports that use parameterized filters either using **composer** or the Dynamic Workload Console, use as value the one reported in the Cognos Report Studio.

To specify parameters that use parameterized filters using the Dynamic Workload Console, perform the following procedure.

1. Select the name of the parameter you want to specify for the report.
2. In the **Value** field, enter the value reported in the Cognos Report Studio. To identify the value proceed as follows:
   a. Open Report Studio.
   b. Open the report to run.
   c. In the Insertable Objects pane, select the filter you want to use. The Properties panel is displayed.
   d. Select the **Ref** value.
   e. Copy this value in the **Value** field.

This example shows how to run the Cognos Employee Training by Year sample report, specifying for the **?pYear?** parameter the value associated with the filter **2004**. The Employee Training by Year sample report is located under `/Samples/Models/Dashboard Objects`. To run the report proceed as follows:

1. In the Insertable Objects pane, select the **2004** filter. The Properties panel is displayed.
2. Select **[go_data_warehouse].[2004]**.
3. Insert **[go_data_warehouse].[2004]** in the **Value** field.
4. Save the job definition.

To specify parameters that use parameterized filters using **composer**, perform the following procedure.

1. Open Report Studio.
2. Open the report to run.
3. In the Insertable Objects pane, select the filter you want to use. The Properties panel is displayed.
4. Select the **Ref** value.

5. Copy this value in the **<jsdlcognos:parametersValues>** attribute. Below an example for the Employee Training by Year sample report specifying for the **?pYear?** parameter the value associated with the filter **2004**:

```
<jsdlcognos:reportGroup>
   <jsdlcognos:ReportPathGroup>
     <jsdlcognos:reportPath>
       Employee Training - Path:/content/folder[@name='Samples']
                               /folder[@name='Models']
                              /folder[@name='Dashboard Objects']
                             /report[@name='Employee Training']
     </jsdlcognos:reportPath>
   </jsdlcognos:ReportPathGroup>
   <jsdlcognos:parametersValues>
        <jsdlcognos:parametersValue>
           key="pYear">[go_data_warehouse].[2004]
        </jsdlcognos:parametersValue>
   </jsdlcognos:parametersValues>
        <jsdlcognos:outputFile>
            C:\Cognos reports\Employee Training
        </jsdlcognos:outputFile>
        <jsdlcognos:formatTypes>HTML</jsdlcognos:formatTypes>
 </jsdlcognos:reportGroup>
```

Scheduling and submitting job streams for Cognos reports

You schedule Tivoli Workload Scheduler jobs for Cognos reports by defining them in job streams.

Distributed Use the Dynamic Workload Console or the command line.

z/OS Use the Dynamic Workload Console or the ISPF application.

After you define a Tivoli Workload Scheduler job for a Cognos report, add it to a job stream with all the necessary scheduling arguments and submit it. After submission, when the job is running (**EXEC** status), you can kill the Tivoli Workload Scheduler job that runs the Cognos report if necessary. This action is converted into a **Cancel** action for the Cognos report.

The agent might become unavailable while the Tivoli Workload Scheduler job running the Cognos report is running. When the agent becomes available again, Tivoli Workload Scheduler starts to monitor the report from where it stopped.

Monitoring Tivoli Workload Scheduler jobs that run Cognos reports

Monitor Tivoli Workload Scheduler jobs that run Cognos reports by using the Dynamic Workload Console, the command line, or the ISPF application.

Distributed You monitor jobs by using the Dynamic Workload Console connected to a distributed engine or by using the **conman** command line.

z/OS You monitor jobs by using the Dynamic Workload Console connected to a z/OS engine or the ISPF application.

For detailed information about:

How to monitor jobs by using the Dynamic Workload Console
See the online help or Monitoring your Objects in the Plan > Monitoring your Workload >Creating a task to monitor jobs in the *Dynamic Workload Console User's Guide*.

How to monitor jobs by using conman

See Managing objects in the plan - conman in *Tivoli Workload Scheduler User's Guide and Reference*.

How to monitor jobs by using the ISPF application

See Monitoring the Workload in *Tivoli Workload Scheduler for z/OS Managing the Workload*.

Mapping between job statuses and Cognos report statuses

Map job status to Cognos report status to understand their processing.

Table 80 table shows how you can map the job status to the Cognos report status based on the return code you find in the job log output.

Table 80. Mapping between job statuses and Cognos report statuses

Cognos report status	Dynamic Workload Console job status	Tivoli Workload Scheduler job status	Tivoli Workload Scheduler for z/OS job status
Executing	Running	EXEC	Executing
Pending	Running	EXEC	Executing
Succeeded	Successful	SUCC	Completed
Failed	Error	ABEND	Error
Canceled	Error	ABEND	Error
Suspended	Running	EXEC	Executing

Analyzing the job log

When a job runs Tivoli Workload Scheduler creates a job log that you can analyze to verify the job status.

Distributed You analyze the job log by using the Dynamic Workload Console or the **conman** command line.

z/OS You analyze the job log by using the Dynamic Workload Console or the ISPF application.

For detailed information about:

How to analyze the job log using the Dynamic Workload Console

See the online help or **Monitoring your Objects in the Plan** > **Monitoring your Workload** > **Creating a task to monitor jobs** in the *Dynamic Workload Console User's Guide*.

How to analyze the job log using conman

See **Managing objects in the plan - conman** > **Conman commands** > **showjobs** in the *Tivoli Workload Scheduler User's Guide and Reference*.

How to analyze the job log using the ISPF application

See Monitoring the Workload in the *Tivoli Workload Scheduler for z/OS Managing the Workload* manual.

Job log output

The Tivoli Workload Scheduler for Cognos report job log and its content.

Purpose

The output of a Tivoli Workload Scheduler job forCognos report shows:

Distributed **Environment:**

- In the first part the JSDL definition you submitted.
- In the second part how the job completed.

See “Sample in a distributed environment.”

z/OS **Environment:**
How the job completed. See “Sample in a z/OS environment” on page 265.

Distributed

Sample in a distributed environment

This example shows the output of a job that run on a dynamic agent that completed successfully:

```
%sj NC125152#JOBS.REPOR1722160684;std
=================================================================
= JOB       : NC125152#JOBS[(0000 02/27/12),(JOBS)].REPOR1722160684
= TASK      : <?xml version="1.0" encoding="UTF-8"?&gt;
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
 xmlns:jsdlcognos="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdlcognos"
 name="COGNOS">
  <jsdl:application name="cognos">
<jsdlcognos:cognos>
 <jsdlcognos:CognosParameters>
   <jsdlcognos:CognosPanel>

.....
.....

</jsdl:jobDefinition>
= TWSRCMAP  :
= AGENT     : NC125152
= Job Number: 1060841359
= Mon Feb 27 17:22:30 CET 2012
=================================================================
AWKCGE050I The IBM Cognos report with path
 "/content/package[@name='tws4apps']/report[@name='Address Report']" started
     running.
AWKCGE051I The IBM Cognos report with path "/content/package[@name='tws4apps']
/report[@name='Address Report']" completed successfully.
AWKCGE053I The IBM Cognos report results were sent by email to
     "gseghetti@it.ibm.com".
AWKCGE052I The IBM Cognos report results were saved in the file
 "C:\Cognos reports\ReportForEuropeBusiness.html".
=================================================================
= Exit Status           : 0
= Elapsed Time (Minutes) : 1
= Mon Feb 27 17:22:44 CET 2012
=================================================================
```

This example shows the output of a job that run on a dynamic agent that completed with errors:

```
%sj NC125152#JOBS.REPOR1726171742;std
=================================================================
= JOB       : NC125152#JOBS[(0000 02/27/12),(JOBS)].REPOR1726171742
= TASK      : <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdl"
 xmlns:jsdlcognos="http://www.ibm.com/xmlns/prod/scheduling/1.0/jsdlcognos"
   name="COGNOS">
```

```
  <jsdl:application name="cognos">
    <jsdlcognos:cognos>
.....
.....


</jsdl:jobDefinition>
= TWSRCMAP  :
= AGENT     : NC125152
= Job Number: 1060841360
= Mon Feb 27 17:26:30 CET 2012
===============================================================
AWKCGE050I The IBM Cognos report with path "/content/package[@name='cognosTime']
/interactiveReport[@name='date and time report']" started running.
AWKCGE056E The IBM Cognos report completed with errors.
===============================================================
= Status Message:  AWKCGE056E The IBM Cognos report completed with errors.
= Exit Status           : -1
= Elapsed Time (Minutes) : 1
= Mon Feb 27 17:26:37 CET 2012
===============================================================
```

z/OS

Sample in a z/OS environment

This example shows the output of a job that run on a dynamic agent that completed successfully:

```
AWKCGE050I The IBM Cognos report with path
 "/content/folder[@name='Samples']/folder[@name='Models']
/package[@name='GO Data Warehouse\
(query)']/folder[@name='Report Studio Report Samples']
/report[@name='Total Revenue by Country']"
started running.
AWKCGE051I The IBM Cognos report with path
 "/content/folder[@name='Samples']/folder[@name='Models']
/package[@name='GO Data Warehouse\
(query)']/folder[@name='Report Studio Report Samples']
/report[@name='Total Revenue by Country']"
completed successfully.
AWKCGE052I The IBM Cognos report results were saved in the file "c:\addrep1.csv".
AWKCGE053I The IBM Cognos report results were sent by email to
   "ri.val@it.ibm.com".
```

This example shows the output of a job that run on a dynamic agent that completed with errors:

```
AWKCGE050I The IBM Cognos report with path
 "/content/package[@name='tws4apps']
/folder[@name='Reports with parameters and prompts']
/interactiveReport[@name='Report 7 with special chars']" started running.
AWKCGE056E The IBM Cognos report completed with errors.
```

Part 8. Managing SAP PI channel jobs

Overview

Define SAP Process Integration (PI) Channel jobs to control communication channels between the Process Integrator and a backend SAP R/3 system.

In a SAP environment, communication channels are used to convert documents. You can use Tivoli Workload Scheduler to control these communication channels by installing the Tivoli Workload Scheduler for Applications SAP PI Channel plug-in and defining SAP PI Channel jobs.

You then schedule SAP PI Channel jobs with Tivoli Workload Scheduler to take full advantage of the Tivoli Workload Scheduler functions to manage jobs.

You can manage these jobs both in a distributed and in a z/OS environment, by selecting the appropriate engine.

Business scenario

A practical business scenario demonstrating how SAP PI Channel jobs can be used to manage communication channels.

A large, multinational manufacturer interacts with business partners, clients, and suppliers in different geographic locations. The computers used by each party differs in hardware, operating system, and more often than not, both. Part of the interactions between these parties include secure data exchange to transmit critical business information. Given the differences in hardware and operating systems, the information is converted to the SAP Intermediate Document (IDoc) format for transferring data related to business transactions. The IDocs are generated in the sending database and is transmitted to the receiver, which in this case, is a SAP R/3 system.

In this scenario, only if files arrive with specific file names, such as `ABC.txt`, `DEF.txt`, and `GHI.txt`, should they be transformed by the process integrator into a single IDoc. To achieve this, the company leverages Event Driven Workload Automation in which an event rule is created that monitors the arrival of files. The action associated with the event rule is to open the sender communication channel. After the conversion to IDoc is complete, the process integrator removes the original source files. The removal of the files is detected by Tivoli Workload Scheduler and a job is submitted to close the sender channel until the next iteration of this same process.

There are several advantages to using Tivoli Workload Scheduler SAP PI Channel Plug-in jobs to control communication channels. For example, they can also be used when the backend SAP R/3 system is under maintenance. A job can be submitted into a job stream to stop the receiving channel and then reactivate it once the maintenance is complete.

Software requirements

You use Tivoli Workload Scheduler Plug-in for SAP PI Channel both in a distributed and in a z/OS environment. You must install the Tivoli Workload Automation product appropriate for your environment, and the SAP PI Channel product. To install SAP PI Channel Plug-in, see "Installing and uninstalling IBM Tivoli Workload Scheduler for Applications Plug-ins" on page 41. In details:

To manage SAP PI Channel jobs in a distributed environment, install:

- Tivoli Workload Scheduler
- Tivoli Workload Scheduler dynamic domain manager or dynamic agent
- SAP PI Channel on the same workstation where you installed the dynamic domain manager or the dynamic agent.
- SAP NetWeaver 2004s XI 3.0 - Support Package 19
- SAP NetWeaver 2004s PI 7.0 - Support Package 12

To manage SAP PI Channel jobs in a z/OS environment, install:

- Tivoli Workload Scheduler for z/OS
- Tivoli Workload Scheduler for z/OS dynamic domain manager, theTivoli Workload Scheduler for z/OS agent, or the dynamic agent.
- SAP PI Channel on the same workstation where you installed the dynamic domain manager, the Tivoli Workload Scheduler for z/OS agent, or the dynamic agent.
- SAP NetWeaver 2004s XI 3.0 - Support Package Stack 19
- SAP NetWeaver 2004s PI 7.0 - Support Package Stack 12

For detailed information about the Tivoli Workload Scheduler supported operating systems, see the System Requirements Document. For more information about SAP NetWeaver, refer to the SAP Help Portal at http://help.sap.com/nw2004/.

Tasks

Using Tivoli Workload Scheduler for Applications Plug-in for SAP PI Channel, you can perform on SAP PI Channel jobs all the tasks you can perform on Tivoli Workload Scheduler jobs by using the Dynamic Workload Console, the command line, or the ISPF interface.

Roles and responsibilities

The user that starts, stops, and verifies the status of SAP PI Channel channels must have the following required J2EE roles:

- **xi_af_channel_admin_display**: to verify the status of communication channels.
- **xi_af_channel_admin_modify**: to perform start and stop actions on communication channels.

Assigning J2EE roles

The user controlling the communication channels must have the required J2EE roles assigned.

Assign the required roles to the user that controls the channel using the J2EE Engine Visual Administrator graphical user interface.

1. Log on to the J2EE Visual Administrator.
2. Click **Server0** > **Services** > **Security Provider**

3. In the right pane, click **sap.com/com.sap.aii.af.app*AdapterFramework**.
4. Select the **Security Roles** page.
5. Assign the user the following roles:
   - `xi_af_channel_admin_display`
   - `xi_af_channel_admin_modify`

Configuring external control

To administrate communication channels using Tivoli Workload Scheduler, you must configure the channel to accept external calls.

To perform actions on SAP PI Channel communication channels such as starting, stopping, and verifying the status of channels, external control must be enabled for the channels to accept HTTP requests.

You can enable external control for communication channels by setting **Control Data** to **External** in the Runtime Workbench Communication Channel Monitoring.

Defining a Tivoli Workload Scheduler job that runs an SAP PI Channel job

Define Tivoli Workload Scheduler jobs that run SAP PI Channel jobs by using either the Dynamic Workload Console or the command line.

In a distributed environment, you define a Tivoli Workload Scheduler job that runs an SAP PI Channel job by using the Dynamic Workload Console connected to a distributed engine or the **composer** command line. In a z/OS environment, you define jobs by using the Dynamic Workload Console connected to a z/OS engine.

Defining a Tivoli Workload Scheduler job that runs an SAP PI Channel job by using composer

You define a Tivoli Workload Scheduler job that runs an SAP PI Channel job by using **composer** if you are connected to a distributed engine.

Purpose

Use the syntax described below to define a Tivoli Workload Scheduler job that runs an SAP PI Channel job.

Syntax

```
$jobs
[workstation#]jobname
   task job_definition [streamlogon username]}
  [description "description"]
  [tasktype tasktype]
  [interactive]
  [rccondsucc "Success Condition"]
  [recovery
      {stop | continue | rerun}
      [after [workstation#]jobname]
      [abendprompt "text"] ]
```

You use the **task** argument specifying the XML syntax for the SAP PI Channel.

For a detailed description of the XML syntax, see "Job definition for SAP PI Channel jobs."

For the job definition, see **Defining objects in the database** > **Defining scheduling objects** > **Job definition** in the *Tivoli Workload Scheduler User's Guide and Reference*.

Job definition for SAP PI Channel jobs

Tivoli Workload Scheduler job definition properties and JSDL examples for running SAP PI Channel jobs.

Table 81 describes the required and optional attributes for SAP PI Channel jobs, together with a description of each attribute.

Table 81. Required and optional attributes for the job definition of SAP PI Channel jobs.

Attribute	Description/value	Required
Host name	Use this section to specify the options for the SAP Process Integration server.	✓
Server connection port	The port number of the SAP Process Integration instance.	✓
Service	Identifies the service of the channel to be administered. You can specify an asterisk (*) to administer more than one channel simultaneously.	
Party	Identifies the party of the channel to be administered. You can specify an asterisk (*) to administer more than one channel simultaneously.	
Channel	Identifies the name of the channel to be administered. You can specify an asterisk (*) to administer more than one channel simultaneously.	✓
User name	The user that controls the channels. This user requires the **`xi_af_channel_admin_display`** and **`xi_af_channel_admin_modify`** roles.	✓
Password	The password of the user. The password is encrypted when the job is created.	✓
Action	Start, stop, verify status.	

The following example shows the job definition of an SAP PI Channel job with all the attributes specified:

```
TWSAGENT#CHANNELTEST
 TASK
    <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.ibm.com/xmlns/prod/scheduling/1.0/
   jsdl" xmlns:jsdlpichannel="http://www.ibm.com/xmlns/prod/scheduling/1.0/
   jsdlpichannel" name="PICHANNEL">
  <jsdl:application name="pichannel">
    <jsdlpichannel:pichannel>
                        <jsdlpichannel:PIChannelParameters>
                                <jsdlpichannel:PIChannelParms>
                                        <jsdlpichannel:ServerInfo>
                                             <jsdlpichannel:HostName>pihost</jsdlpichannel:HostName>
                                             <jsdlpichannel:PortNumber>50000</jsdlpichannel:PortNumber>
                                        </jsdlpichannel:ServerInfo>
                                        <jsdlpichannel:ChannelInfo>
                                             <jsdlpichannel:ServiceName>*</jsdlpichannel:ServiceName>
                                             <jsdlpichannel:PartyName>*</jsdlpichannel:PartyName>
                                             <jsdlpichannel:ChannelName>TESTCHANNEL1</jsdlpichannel:ChannelName>
                                        </jsdlpichannel:ChannelInfo>
                                        <jsdlpichannel:UserInfo>
                                             <jsdlpichannel:UserName>TWSADMIN</jsdlpichannel:UserName>
                                             <jsdlpichannel:password>
    {aes}VlHkyc5ufaC6nMRepctNUbZ1exnDF5zUl+9baDGWgos=</jsdlpichannel:password>
                                        </jsdlpichannel:UserInfo>
                                        <jsdlpichannel:ActionInfo>
                                                <jsdlpichannel:StartAction/>
                                        </jsdlpichannel:ActionInfo>
                                </jsdlpichannel:PIChannelParms>
                        </jsdlpichannel:PIChannelParameters>
                </jsdlpichannel:pichannel>
  </jsdl:application>
</jsdl:jobDefinition>
 RECOVERY STOP
```

Defining Tivoli Workload Scheduler jobs to run SAP PI Channel jobs by using the Dynamic Workload Console

You define jobs by using the Dynamic Workload Console when you are working in either a distributed environment or in a z/OS environment.

To define a job that runs an SAP PI Channel job by using the Dynamic Workload Console, perform the following procedure.

1. In the console navigation tree, expand **Workload** > **Design** and click **Create Workload Definitions**
2. Specify an engine name, either distributed or z/OS. The Workload Designer opens.
3. In the Working List panel, select the SAP PI Channel job definition.

   z/OS **Environment**

   New > **ERP** > **SAP PI Channel**

   Distributed **Environment**

   New > **Job Definition** > **ERP** > **SAP PI Channel**

   The properties of the job are displayed in the right-hand panel for editing.
4. In the properties panel, specify the attributes for the job definition you are creating. You can find detailed information about all the attributes in the contextual help available on each panel.
5. Click **Save** to save the job definition in the database.

You can now proceed to adding the job to a job stream and submitting it to run.

Submitting Tivoli Workload Scheduler job streams for SAP PI Channel jobs

Submitting Tivoli Workload Scheduler forSAP PI Channel jobs.

After you define a Tivoli Workload Scheduler for SAP PI Channel job, you add it to a job stream with all the necessary scheduling arguments and submit it. After submission you can kill the Tivoli Workload Scheduler for SAP PI Channel job if necessary, this action is converted in a **Stop** action for the SAP PI Channel job.

If the Tivoli Workload Scheduler agent is not available when you submit the Tivoli Workload Scheduler for SAP PI Channel job or when the job is running, Tivoli Workload Scheduler collects the job log when the agent restarts and assigns the

Error or **ABEND** status to the Tivoli Workload Scheduler for SAP PI Channel job independently of the status of the SAP PI Channel job.

Monitoring SAP PI Channel jobs

Monitor jobs by using the Dynamic Workload Console, the command line, or the ISPF application.

In a distributed environment, you monitor jobs by using the Dynamic Workload Console connected to a distributed engine or by using the **conman** command line. In a z/OS environment, you monitor jobs by using the Dynamic Workload Console connected to a z/OS engine or the ISPF application.

For detailed information about:

How to monitor jobs by using the Dynamic Workload Console
See the online help or Monitoring your Objects in the Plan > Monitoring your Workload >Creating a task to monitor jobs in the *Dynamic Workload Console User's Guide*.

How to monitor jobs by using conman
See Managing objects in the plan - conman in *Tivoli Workload Scheduler User's Guide and Reference*.

How to monitor jobs by using the ISPF application
See Monitoring the Workload in *Tivoli Workload Scheduler for z/OS Managing the Workload*.

Mapping between Tivoli Workload Scheduler and SAP PI Channel job statuses

Map Tivoli Workload Scheduler job status to SAP PI Channel job status to understand their processing.

Table 82 shows how you can map the Tivoli Workload Scheduler job status to the SAP PI Channel job status based on the return code you find in the job log output.

Table 82. Mapping between Tivoli Workload Scheduler and SAP PI Channel job statuses

SAP PI Channel Communication Status	Dynamic Workload Console Job Status	Tivoli Workload Scheduler Job Status	Tivoli Workload Scheduler for z/OS Job Status
Green	Running	EXEC	Executing
Green	Successful	SUCC	Completed
Red	Error	ABEND	Error
Yellow	Error	ABEND	Error
Grey	Error	ABEND	Error
Not Available	Error	FAILED	Error

Analyzing the job log

When a job runs Tivoli Workload Scheduler creates a job log that you can analyze to verify the job status.

Distributed You analyze the job log by using the Dynamic Workload Console or the **conman** command line.

z/OS You analyze the job log by using the Dynamic Workload Console or the ISPF application.

For detailed information about:

How to analyze the job log using the Dynamic Workload Console
See the online help or **Monitoring your Objects in the Plan** > **Monitoring your Workload** > **Creating a task to monitor jobs** in the *Dynamic Workload Console User's Guide*.

How to analyze the job log using conman
See **Managing objects in the plan - conman** > **Conman commands** > **showjobs** in the *Tivoli Workload Scheduler User's Guide and Reference*.

How to analyze the job log using the ISPF application
See Monitoring the Workload in the *Tivoli Workload Scheduler for z/OS Managing the Workload* manual.

Job log output

The Tivoli Workload Scheduler for SAP PI Channel job log and its content.

Purpose

The output of a Tivoli Workload Scheduler for SAP PI Channel job relays valuable information to perform problem determination such as:

- If the user has the required roles.
- If the communication channels are enabled for external control.
- If the user name or password are valid.
- If the hostname is resolvable.
- If the hostname is resolvable but not listening on the indicated port.
- If the channel exists.
- If the hostname and port are working correctly, but there is no process integration running.

Sample

This example shows the output of a job where the hostname was resolved successfully, the port is listening, the process integration is running, the channel exists and is enabled for external control, and the start action is being performed:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ChannelAdminResult xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://pihost:50000/AdapterFramework/channelAdmin/
      ChannelAdmin.xsd">
        <Channels>
                <Channel>
                        <Party></Party>
                        <Service></Service>
                        <ChannelName>TESTCHANNEL1</ChannelName>
                        <ChannelID>f750195443af39b2be83dd5c3686983d</ChannelID>
                        <ActivationState>STARTED</ActivationState>
                        <ChannelState>OK</ChannelState>
                </Channel>
        </Channels>
</ChannelAdminResult>

Exit Status  : 0
```

Part 9. Common serviceability

The return code mapping feature

The return code mapping feature provides a standard way of mapping messages into return code values. You can also customize the return code mapping. This feature is available for the following access methods:
- Oracle E-Business Suite
- PeopleSoft
- SAP R/3

The return code mapping feature provides more granularity when defining the success or failure policies of jobs and improved flexibility in controlling job execution flows based on execution results. Job return code mapping provides the following capabilities:
- Users can define a job final status (successful or failed) based on a condition on the return code of the execution of the program or script of the job.
- The return code can be provided also to the recovery job that is associated with it in the job definition. This causes the recovery job to perform different processing based on the return code.

Creating a return code mapping file

You can create a return code mapping file to customize your own return codes with respect to certain conditions that might affect a job when it runs. Use this file to set the success condition of the job, which Tivoli Workload Scheduler uses to assess if the job completes successfully or in error. The return code is sent to Tivoli Workload Scheduler in the form of a `%RC nnnn` message. If this message is received, the job state is updated accordingly.

Each method has its own set of files to map the messages into return code values. The mapping files can be either global or local for a workstation.

Return code mapping files that are specific to a workstation are named according to the following scheme:

```
TWS_home/methods/rcm/accessmethod-type-workstation.rcm
```

Global mapping files have a file name according to the following scheme:

```
TWS_home/methods/rcm/accessmethod-type.rcm
```

For the Oracle E-Business Suite and PeopleSoft access methods, *type* is always equal to `rcmap`. For the SAP R/3 access method, *type* is as described in "Return code mapping file names for r3batch" on page 280.

Syntax

Use the following syntax to create the return code mapping file:

```
[#] "pattern1" "pattern2"..."patternn" = RC value
```

Parameters:

Optional comment. All the lines starting with this symbol (#) are not used for mapping.

pattern*n*

Pattern strings delimited by quotation marks (“ and ”). If you use only one pattern string, you can omit the quotation marks. If the pattern string contains a quotation marks character, then it must be escaped by backslash (\). The string can contain the following wildcards and special characters:

Asterisk (*)

Matches an arbitrary number of characters.

Question mark (?)

Matches a single character.

Backslash (\)

Escape character.

RC value

The return code value. This value is sent by the method to Tivoli Workload Scheduler by a `%RC nnnn` message.

Examples

The following is an example of a return code mapping file. The line numbers in bold do not belong to the file, but are shown for reference:

```
1.   # This is an RC mapping file for joblog.
2.
3.   "User * missing "  = 102
4.   "\*\*\*"  = 103
5.   "User \
6.   * \
7.   missing"  = 102
```

In this example:

- Line **1** is a comment and is not used for mapping.
- Line **2** is blank and is ignored. All blanks preceding or following a pattern string are ignored, as well as those between the equals sign and the return code value.
- Line **3** matches every message starting with the string *User* and ending with the string *missing*.
- Line **4** matches every message starting with three asterisks (*) followed by a blank. When you use the asterisk in this way and not as a wildcard, you must escape it with a backslash.
- Lines **5** through **7** contain a pattern taking several lines. It matches the same messages as the pattern of line **3**.

Considerations

Note the following facts:

- The order of the pattern lines is important because the first matching pattern line is used to build the return code value.
- Empty pattern strings (“”) are ignored by the pattern matching procedure.

For example, the following is a valid pattern sequence. The first line is more restrictive than the second line.

```
"625" "User * missing" = 104
"" "User * missing" = 102
```

The following pattern sequence is formally valid, but the second pattern line is never used. Because the first line is more general, it is always matched first.

```
"" "User * missing" = 102
"625" "User * missing" = 104
```

Return code mapping for psagent and MCMAGENT

For the Oracle E-Business Suite and PeopleSoft access method, you can write return code mapping files associating the internal states listed in Table 83 and Table 84.

When no return code mapping files are defined, or when a string returned by the access method does not satisfy any of the matching patterns of the mapping file, the access method uses the respective standard return codes listed in the tables.

Table 83. Job states and return codes for the Oracle E-Business Suite access method

MCMAGENT job state	MCMAGENT return code
"RUNNING"	1
"PENDING"	2
"COMPLETE"	3
"INACTIVE"	4
"NORMAL"	5
"TERMINATING"	6
"PAUSED"	7
"RESUMING"	8
"STANDBY"	9
"SCHEDULED"	10
"WAITING"	11
"WARNING"	13
"ERROR"	14
"TERMINATED"	15
"DELETED"	16
"ON HOLD"	17
"DISABLED"	18
"NO MANAGER"	19
"SUSPENDED"	20
"TEST FAILURE"	21
"NOT FOUND"	22

Table 84. Job states and return codes for the PeopleSoft access method

psagent job state	psagent return code
"CANCEL"	1
"DELETE"	2
"ERROR"	3
"HOLD"	4
"QUEUED"	5
"INITIATED"	6
"PROCESSING"	7
"CANCELED"	8
"SUCCESS"	9

Table 84. Job states and return codes for the PeopleSoft access method (continued)

psagent job state	psagent return code
"NO SUCCESSPOSTED"	10
"POSTED"	11
"NOT POSTED"	12
"RESEND"	13
"POSTING"	14
"GENERATED"	15

Examples

The extended agent for Oracle E-Business Suite returns the TERMINATED string. The resulting return code depends on the following conditions:

- No return code mapping file exists. In this case, the method sends a return code value of 15, as listed in Table 83 on page 277.
- A return code mapping file exists and is:

  ```
  "NORMAL"=0
  "TERMINATED","DELETED"=1
  ```

  In this case, the access method finds the matching pattern and sends a return code value of 1 as listed in Table 83 on page 277.
- The return code mapping file is:

  ```
  "NORMAL"=0
  "ERROR","DELETED"=1
  ```

  In this case, the access method does not find the matching pattern and sends a return code value of 15 as listed in Table 83 on page 277.

Return code mapping for r3batch

Using return code mapping with r3batch can be useful in overcoming differences in the return code mechanisms of R/3, which returns a mixture of messages and numbers, and of Tivoli Workload Scheduler, which handles exclusively numeric return codes. By customizing the return code mapping files listed in “Return code mapping file names for r3batch” on page 280, you can map messages from R/3 logs, spool lists, and exceptions from RFC function modules into return code values that Tivoli Workload Scheduler can handle.

Note that when you do not use this feature, r3batch does not send any return codes to Tivoli Workload Scheduler. In this case, Tivoli Workload Scheduler displays only the r3batch exit code, which cannot be used to set up rccondsucc conditions.

The return code mapping mechanism works as follows:

1. r3batch reads the output retrieved from the R/3 system (R/3 job log, process chain log, spool list, and so on appended to the stdlist of the related Tivoli Workload Scheduler job).
2. Following your specifications in the rcm files, the R/3 return messages or codes are mapped into your custom return codes and passed on to Tivoli Workload Scheduler.
3. These return codes are used together with the rccondsucc keyword set in the extended agent job definition and handled accordingly. Return code mapping is meaningful only if you use the return codes to write the expressions that

determine job completion. Conversely, in the case of this extended agent, the use of rccondsucc is significant only if Tivoli Workload Scheduler gets return codes (not exit codes) from the access method.

To use the return code mapping feature:

- Leave the value of the rcmap option as ON (this is the default).
- Depending on which R/3 logs you want r3batch to read and map, leave the default settings of the retrieve_joblog, retrieve_pchainlog, and retrieve_spoollist options as ON and manually create the corresponding rcm files.
- If you want to map messages from the R/3 syslog, set the log_r3syslog option to ON and manually create the corresponding rcm file.

When setting up your return code mapping for r3batch, consider the following:

- You can define any return code numbers for your use because there are no reserved return codes for the access method or for Tivoli Workload Scheduler.
- Mapping files are scanned sequentially: the first match found performs the corresponding mapping. When you define a mapping file, write the most restrictive strings first.
- When you define a mapping file, remember that the R/3 log messages are read in their entirety. If you want to map only a part of the entry, you must use the wildcard characters.
- If two lines match two different patterns, then the return code is set to the higher value. In general the return code is set to the highest value among the ones yielded by the matched patterns. This is shown in the following example:

  The job log returned after job PAYT410 has run is:

  ```
  *** ERROR 778 ***  EEWO0778E Failed to modify the job PAYT410 with job id
  ***       05710310.

  *** ERROR 176 ***  EEWO0176E Failed to add step 1.

  *** ERROR 552 ***  EEWO0552E The R/3 job scheduling system has found an
  ***       error for user name * and job name PAYT410. Please check R/3
  ***       syslog.

  *** ERROR 118 ***  EEWO0118E Execution terminated. Could not create and
  ***       start an instance of the R/3 batch job.
  ERROR LEVEL=118
  ```

  and the system log contains the following line:

  ```
  |011:05:12|MAESTRO|SAPMSSY1|EFT|> Step 1 contains illegal values
  ```

  The r3batch-joblog.rcm file contains the following matching line:

  ```
  "118""*"=100
  ```

  while the r3batch-syslog.rcm file contains the following matching line:

  ```
  "*MAESTRO*Step 1 contains illegal values"=9999
  ```

  In this case, the return code sent back to Tivoli Workload Scheduler is 9999 because it is the higher of the two matching patterns.
- If no matching takes place, no return code is sent to Tivoli Workload Scheduler.

Return code mapping file names for r3batch

`r3batch` uses the global return code mapping files listed below. You must manually create the `rcm` directory under *TWS_home*/`methods` and the mapping files you want to implement.

TWS_home/**methods/rcm/r3batch-joblog.rcm**

Maps messages from the R/3 job log of a job into return code values. If this file is not present, the messages in the joblog are ignored.

The format of the mapping file is:

```
 message_text_pattern
[program_pattern[message_number_pattern[message_id_pattern]]]=RCvalue
```

where *program_pattern* is the external program that produced the output shown in the job log and *message_id_pattern* is the message class. For example, the following line appended in the job log:

```
04/26/2005 10:08:04 00
    550Step 001 started (program BTCTEST, variant GIULIO, user name TWSDEV)
```

will match the following pattern line in *TWS_home*/`methods/rcm/r3batch-joblog.rcm`:

```
"*Step*" "*" "550" "*"=5
```

because:

message_text_pattern
"Step 001 started (program BTCTEST, variant GIULIO, user name TWSDEV)"

program_pattern
"*"

message_number_pattern
"550"

message_id_pattern
"*"

TWS_home/**methods/rcm/r3batch-pchainlog.rcm**

Maps messages from the protocol of a Process Chain into return code values. If this file is not present, the messages in the protocol are ignored.

The format of the mapping file is:

```
 message_number_pattern
[message_id_pattern[message_variable1[message_variable2
[message_variable3[message_variable4[message_type]]]]]]=RCvalue
```

TWS_home/**methods/rcm/r3batch-spoollist.rcm**

Maps messages in the job spool list of an R/3 job into return code values. If this file is not present, the messages in the spool list are ignored.

The format of the mapping file is:

```
spool_list_row_pattern=RCvalue
```

TWS_home/**methods/rcm/r3batch-syslog.rcm**

Maps messages in the syslog of an R/3 system into return code values. The R/3 system log should be checked only when R/3 returns the generic 552 error to `r3batch`.

If this file is not present, the messages in the system log are ignored.

The format of the mapping file is:

```
system_log_row_pattern=RCvalue
```

If you plan to map system log messages, be sure to set the `log_r3syslog` option of `r3batch` to `ON` (the default is `OFF`).

TWS_home/**methods/rcm/r3batch-msgrc.rcm**
Maps ABAP exceptions and BAPI return codes of RFC function modules into return code values. If this file is not present, the mapping is done using a hardcoded table.

The format of the mapping file is:

```
message_number=RCvalue
```

`message_number` is the error message number. The last message number is always used. That is, if two error messages are generated, only the second one is checked against the mapping file.

Mapping return codes for intercepted jobs

To set up return code mapping for intercepted jobs, after defining the appropriate return code conditions in the `r3batch-joblog.rcm` file, do the following:

1. Create a customized template file named *`TWS_home`*`/methods/r3batch_icp/rctemplate.jdf` containing the following:

   ```
   alias;rccondsucc "Success Condition"
   ```

   where the "Success Condition" must match a condition saved in the `rcm` file.
2. Modify the *`TWS_home`*`/methods/r3batch_icp/`*`XANAME`*`_r3batch.icp` referring to the `jdf` file you created as follows:

   ```
   client job_mask user_mask rctemplate
   ```

   Tivoli Workload Scheduler manages the intercepted R/3 job as a docommand job with all the options specified in the customized `jdf` file. You can check if your intercepted job is correctly submitted by reading the job_interceptor joblog.

Configuring the tracing utility

This chapter describes how to configure the trace utility for all the access methods. Tivoli Workload Scheduler for Applications logs all the processing information in the *`TWS_home`*`/methods/`*`accessmethod`*`.properties` configuration file.

Note: If you delete this file accidentally, Tivoli Workload Scheduler for Applications creates a new file with all the default values and contains the following comment:

```
# This file was automatically created using the default values.
```

Customizing the .properties file

Depending on the access method you are working with, customize the trace parameters in the following properties files:

MCMAGENT.properties
For the Oracle E-Business Suite access method.

psagent.properties
For the PeopleSoft access method.

r3batch.properties, r3evmon.properties
For the SAP R/3 access method.

With this access method, you can also specify debug and trace parameters in the single job definitions. See R/3 Job Definition and "Task string to define SAP jobs" on page 115.

mvsjes.properties, mvsopc.properties
For the z/OS access method, depending on the scheduler with which you are working.

For each .properties file you can customize the following parameters:

*accessmethod***.trace.tracers.level**
Specify the level of tracing you want to set. Possible values are:

DEBUG_MIN
Only error messages are written in the trace file. This is the default.

DEBUG_MID
Informational messages and warnings are also written in the trace file.

DEBUG_MAX
A most verbose debug output is written in the trace file.

The value you set in the .properties file applies to all the jobs of the corresponding access method. To specify a different trace setting for a particular job, specify the following option in the job definition:

```
-tracelvl=(1|2|3)
```

where:

- `1 = DEBUG_MIN`
- `2 = DEBUG_MID`
- `3 = DEBUG_MAX`

| **Note:** When making changes to the trace level setting, the changes are
| effective immediately after saving the .properties file. No restart is
| required.

*accessmethod***.trace.handlers.traceFile.fileDir**
Specifies the path where the trace file is created. Depending on the access method, the default is:

SAP R/3
`TWS_home``/methods/traces`

All other access methods
`TWS_home``/methods`

Ensure that the new path you specify has already been created as a fully qualified path with write permissions.

Traces are written in XML format. Always use slashes (or backslashes) when you specify a new path, even if you are working on Windows operating systems.

The trace files give information about the method execution to the desired level of detail. The minimum trace level is always on, to guarantee a First-Failure Data Capture (FFDC) ability. The trace file name is:

trace-MCMAGENT.log
For the Oracle E-Business Suite access method.

trace-psagent.log
For the PeopleSoft access method.

trace-r3batch.log, trace-*XAname*-r3evmon.log
For the SAP R/3 access method.

trace-mvsjes.log, trace-mvsopc.log
For the z/OS access method. Depending on the scheduler with which you are working.

***accessmethod*.trace.tracers.logging**
Specifies to enable or disable the trace utility. Possible values are:

true To enable the trace utility. This is the default value.

false To disable the trace utility. If you set this parameter to **false**, no traces are written in the trace-*accessmethod*.log file even if there are problems.

r3batch.trace.handlers.traceFile.maxFiles
The maximum number of trace files that are created before the oldest one is deleted. If this parameter is set to 1, the current trace file is never replaced and can grow without limit.

r3batch.trace.handlers.traceFile.maxFileSize
The maximum size (in bytes) that the trace file can reach before it is renamed and a new trace file is created. This parameter is valid only if the r3batch.trace.handlers.traceFile.maxFiles is set to a value greater than 1.

Configuration file example for the SAP access method

The following `r3batch.properties` file is an example of a configuration file for the SAP access method with the following characteristics:

- The level of tracing set is DEBUG_MID. This means that not only error messages but also informational messages and warnings are written in the trace file.
- The trace file is created in the `/home/maestro/methods` directory.
- The tracing process creates three trace files, whose maximum size can be 10 MB.

```
r3batch.organization=IBM
r3batch.product=TWS4APPS
r3batch.component=R3BATCH
r3batch.trace.tracers.level=DEBUG_MID
r3batch.trace.tracers.listenerNames=r3batch.trace.handlers.traceFile
r3batch.trace.tracers.logging=true
r3batch.trace.handlers.traceFile.fileDir=/home/maestro/methods
r3batch.trace.handlers.traceFile.formatterName=r3batch.trace.formatter
r3batch.trace.handlers.traceFile.maxFileSize=104805100
r3batch.trace.handlers.traceFile.maxFiles=3
```

Troubleshooting

This chapter describes a number of problems that might occur when installing or using Tivoli Workload Scheduler for Applications access methods or plug-ins.

Analyzing return codes for silent installation

Check the error and warning messages issued by Tivoli Workload Scheduler for Applications and IBM Tivoli Workload Scheduler for Applications Plug-ins, during the silent installation process.

This section lists the **errors** and the **warnings** messages returned by InstallAnywhere during the silent installation process.

The **errors** and **warnings** are organized into two tables:

- Default InstallAnywhere error messages, Table 85.
- Tivoli Workload Scheduler for Applications error messages, Table 86 on page 285.

When running the installation in silent mode, the messages are not displayed, but are written in the silent installation log files listed in Table 8 on page 45.

If the response file you specify in the command line does not exist or the file name is incorrect, the silent installation process does not write the log files. To have the correct return code for the silent installation process, issue:

```
start /w <silent command>
```

where:<*silent command*> is the command you launch to perform silent installation.

You can check the error codes found in the installation log files during the silent installation process, with the codes in the following tables to obtain the specific description of the error message. Table 85 shows the default InstallAnywhere error messages written in the log files during the silent installation execution.

Table 85. Default InstallAnywhere error messages

Error Code	Description
0	Success: The installation completed successfully without any warnings or errors.
1	The installation completed successfully, but one or more of the actions from the installation sequence caused a warning or a non-fatal error.
8	The silent installation failed because there is an error in one or more installation steps.
-1	One or more of the actions from the installation sequence caused a fatal error.
1000	The installation was canceled by the user.
1001	The installation includes an invalid command-line option.
2000	Unhandled error.
2001	The installation failed the authorization check. It might indicate an expired version.
2002	The installation failed a rules check. A rule placed on the installer itself failed.
2003	An unresolved dependency in silent mode caused the installer to exit.
2004	The installation failed because not enough disk space was detected while running the Install action.
2005	The installation failed while trying to install on a Windows 64-bit system, because the installation does not include support for Windows 64-bit systems.
2006	The installation failed because it was launched in a UI mode that is not supported by this installer.
3000	Unhandled error specific to a launcher.
3001	The installation failed due to an error specific to the lax.main.class property.
3002	The installation failed due to an error specific to the lax.main.method property.

Table 85. Default InstallAnywhere error messages (continued)

Error Code	Description
3003	The installation was unable to access the method specified in the lax.main.method property.
3004	The installation failed due to an exception error caused by the lax.main.method property.
3005	The installation failed because no value was assigned to the lax.application.name property.
3006	The installation was unable to access the value assigned to the lax.nl.java.launcher.main.class property.
3007	The installation failed due to an error specific to the lax.nl.java.launcher.main.class property.
3008	The installation failed due to an error specific to the lax.nl.java.launcher.main.method property.
3009	The installation was unable to access the method specified in the lax.nl.launcher.java.main.method property.
4000	A Java executable could not be found at the directory specified by the java.home system property.
4001	An incorrect path to the installer jar caused the relauncher to launch incorrectly.

Table 86 shows the error messages issued during Tivoli Workload Scheduler for Applications silent installation of the methods or the plug-ins.

Table 86. InstallAnywhere error messages for methods and plug-ins installation

Error Code	Description
11	The required parameter does not contain a value.
12	The file specified in response file does not exist.
13	The file specified for peoplesoft method installation, is not a *psjoar.jar* file type.
14	You did not specify a feature (methods or plug-ins) to install in the response file.
15	The installation process cannot find a Tivoli Workload Automation instance on this system.
16	You cannot perform the action that you specified on the selected instance.
17	You are performing the installation on a workstation that does not have enough disk space.
18	The path you specified does not contain a valid installation of Tivoli Workload Scheduler.
19	The operating system, where you are performing installation, is not supported.
20	The backup of previous Tivoli Workload Scheduler for Applications installation files failed.
21	The installation process is unable to install the plug-in.
22	The installation process is unable to uninstall the plug-ins.

Known installation problems

Table 87 lists known installation problems.

Table 87. Known installation problems

Area	Item
Installation of the product	**Symptom**: The silent installation of PeopleSoft method fails with exit code 2 in the installation log files listed in Table 1 on page 10.
	Cause and Solution: This is because the PeopleSoft file `psjoa.jar` is not in the directory specified in the option *TWSAPPS_PS_FILE_NAME* of the response file. You must: 1. Copy the file `psjoa.jar` to the directory specified in the option *TWSAPPS_PS_FILE_NAME* of the response file. 2. Start again the silent installation.

Installation of the product	**Symptom**: The installation of Tivoli Workload Scheduler for Applications and IBM Tivoli Workload Scheduler for Applications Plug-ins fails with the message:
	`WARNING:/tmp does not have enough disk space!`
	`Attempting to use / for install base and tmp dir`

	Cause and Solution: If the temporary directory does not have enough space, you must redirect the installation process to another temporary directory, set the InstallAnywhere variable *IATEMPDIR*: **Windows operating systems** 1. set IATEMPDIR=<new_temp_dir> 2. Start the installation. **UNIX operating systems** 1. export IATEMPDIR=<new_temp_dir> 2. Start the installation.
Uninstallation of the product	**Symptom**: You uninstalled the IBM Tivoli Workload Scheduler for Applications Plug-ins version 8.6 using the Tivoli Workload Scheduler for Applications wizard. The wizard panel show the message that uninstallation completes successfully but the uninstallation logs files show the error that the uninstallation process is unable to remove one or more locked file in the installation directory. When you try to reinstall same IBM Tivoli Workload Scheduler for Applications Plug-ins, installation fails.
	Cause and Solution: This is because of InstallAnywhere continues the uninstallation process even if an exceptions is caught. To be able to complete uninstallation and reinstall IBM Tivoli Workload Scheduler for Applications Plug-ins successfully, you must: 1. Remove manually one or more the locked files at the end of uninstallation process. 2. Start the IBM Tivoli Workload Scheduler for Applications Plug-ins installation.

Other known problems

Table 88 lists miscellaneous troubleshooting problems.

Table 88. Miscellaneous troubleshooting items

Area	Item
`r3batch` and `r3event`: output contains unreadable characters	**Symptom**: When you enter the `r3batch` and `r3event` commands interactively (for example, to export an SAP calendar) the output is returned in UTF-8 format.
	Solution: To resolve this problem, you can either use a shell that supports the UTF-8 code page or redirect the output to a file and open it with a text editor that supports the UTF-8 format.

Table 88. Miscellaneous troubleshooting items (continued)

Area	Item
`r3batch`: SAP jobs contain quotation marks (") or reverse quotes (`)	**Symptoms**: SAP jobs whose names contain quotation marks or reverse quotes are not displayed in the pick list of the Dynamic Workload Console. -OR- You have a Tivoli Workload Scheduler job that tries to submit an SAP job whose name contains quotation marks, but it abends with an error. The following message might be displayed: `EEWO0439E The required options are not specified either in the global or in the local options file.`
	Solution: In your SAP system, make a copy of the SAP job and assign it a name that does not contain quotation marks or reverse quotes.
`r3batch`: SAP job containing Arabic characters.	**Symptom**: An SAP job abends when the job contains Arabic characters.
	Solution: If you run an SAP job that contains Arabic characters, you must set the local codepage of the agent workstation hosting the r3batch access method to the Arabic codepage. Refer to the **`twsmeth_cp`** keyword in the common options file, “Defining the common options” on page 94.

`r3batch`: error messages submitting a job on dynamic agents.	**Symptom**: When working with dynamic workstations and performing actions such as: displaying a process chain, restarting a process chain, or retrieving the spool list, the following messages might be displayed from the Dynamic Workload Console:
	`EEWO0439E The required options are not specified either in the global or in the local options file.`
	`EEWO1065W The environment variable UNISON_JOB is not set. The process chain cannot be restarted.`

	Solution: These messages might indicate that the requested action is not supported on dynamic workstations. Refer to the *Tivoli Workload Scheduler for Applications Release Notes* for more information about Tivoli Workload Scheduler for Applications features and minimum required versions for compatibility.
`r3batch`: r3batch hangs when performing actions from the Dynamic Workload Console.	**Symptoms**: r3batch hangs when performing actions from the Dynamic Workload Console such selecting from a pick list, submitting a job, or similar actions that require connection to the SAP system. The Tivoli Workload Scheduler joblog might also contain multiple "Timer expired" messages.
	Solution: This problem is caused by the Tivoli Workload Scheduler logging and tracing component. There are two possible solutions: • Deactivate the tracing utility as described in the following technote: http://www.ibm.com/support/docview.wss?uid=swg21503284. OR • Modify the `r3batch.properties` files. Locate the**`r3batch.trace.handlers.traceFile.MPFileSemKeyproperties`** setting, and then either comment this property setting out or use a different value. Choose any numeric value and retry the operation.
`r3batch`: Submit same process chain in parallel fails.	**Symptom**: The SAP system returns an error message RFC_ERROR_SYSTEM_FAILURE when starting an SAP process chain.
	Solution: Verify if the corrections stated in SAP note 1723482 are applied to your SAP Business Warehouse system or avoid running the same process chain more than once simultaneously.

Table 88. Miscellaneous troubleshooting items (continued)

Area	Item
r3batch: When you restart the process of a subchain, the status of the original process chain is not changed to active	**Symptom**: When you restart the process of a subchain, the status of the original process chain is not changed to active.
	Solution: Refer to SAP Note 1075876.
`r3batch`: Refresh an SAP process chain after a kill action on a running job instance.	**Symptom**: If you perform a kill action on a Tivoli Workload Scheduler job instance running on a dynamic workstation which monitors an SAP process chain, and then subsequently perform a Refresh operation on this job, the job fails.
	Solution: You cannot perform a Refresh operation after having performed a kill action on a Tivoli Workload Scheduler job instance running on a dynamic workstation which monitors an SAP process chain. Verify the status of the SAP process chain on the SAP system, and then set the Tivoli Workload Scheduler job status accordingly.
`r3batch`: Wrong characters are displayed in the Criteria Manager profile description.	**Symptom**: Special characters such as, < (less than), > (greater than), or the ' (apostrophe) specified in the **Description** field of the **Create criteria profile** dialog are displayed incorrectly.
	Solution: Avoid using special characters in the **Description** field when creating a new criteria profile.
`r3evmon`: monitoring events is not started, stopped, or performed	**Symptom**: You cannot start or stop event monitoring, or event monitoring is not performed.
	Solution: Ensure that *TWSuser* is the owner of the following files, and that the user has read and write permissions: • `/`*`TWS_home`*`/pids/`*`XAname`*`_r3evmon.pid` • `/`*`TWS_home`*`/EIF/`*`XAname`*`_r3evmoncache.dat` • `/`*`TWS_home`*`/EIF/`*`XAname`*`_r3evmoneif.conf` • `/`*`TWS_home`*`/methods/r3evmon_cfg/`*`XAname`*`_r3evmon.cfg` • `/`*`TWS_home`*`/methods/r3evmon_cfg/`*`XAname`*`_r3idocmon.cfg` • `/`*`TWS_home`*`/methods/r3evmon_cfg/`*`XAname`*`_r3xalmon.cfg` • `/`*`TWS_home`*`/methods/r3evmon_cfg/`*`XAname`*`_r3evmon.lck`
`r3batch`: monitoring SAP events is not performed	**Symptom**: The SAP event on which the event rule is based is neither monitored nor committed.
	Solution: Ensure that the extended agent workstation you specified in the SAP event definition exists. When you define an SAP event within an event rule, no check on the extended agent workstation is made: if the workstation does not exist, the event rule is saved and activated but it will never be resolved.
`r3batch`: monitoring SAP events is not performed	**Symptom**: With XBP 3.0, the SAP event is raised but Tivoli Workload Scheduler is not notified and therefore does not act as expected.
	Solution: Ensure that the SAP event was not excluded from logging in the SAP event history table.

Table 88. Miscellaneous troubleshooting items (continued)

Area	Item
`r3batch`: monitoring SAP events is not performed	**Symptom**: The SAP events on which the event rule is based are not monitored nor committed.

	Solution: The SAP events being monitored are listed in the following file:
	`TWS_home/monconf/XAname_r3evmon.cfg`
	where *XAname* is the name of the SAP extended agent workstation.
	Check that the file is updated and contains the current monitoring plan. The SAP events are indicated by the following keyword (one for each SAP event on the same extended agent):
	`!R3EVENT SAP_event_name_lengthSAP_event_name[SAP_event_parm_lengthSAP_event_parm]`
	where:
	SAP_event_name_length
	The length of the SAP event name to monitor, in the format *nnnn*. For example, `0008`, if the event name is `SAP_TEST`.
	SAP_event_name
	The name of the SAP event to monitor.
	SAP_event_parm_length
	The length of the parameter associated with the SAP event to monitor, if any. The format is *nnnn*. For example, `0007`, if the event name is `SAP_PAR`.
	SAP_event_parm
	The parameter associated with the SAP event to monitor, if any. This value is optional, but omitting it identifies an SAP event with no parameter associated. For details about how the events are matched between `r3evmon.cfg` and the SAP system, see "SAP events matching criteria" on page 187.
	For each configuration file, an `r3evmon` process is started to monitor the SAP events listed. To start an `r3evmon` monitoring process for a specific extended agent workstation, enter the following command.
	Note:
	1. For UNIX only, `r3evmon` must be entered by the owner of the Tivoli Workload Scheduler installation:
	2. If you run `r3evmon` from a Windows DOS shell, the command prompt is not returned until the process completes.
	`r3evmon -t SEM -c XAname -- "[-EIFSRV EIF_server -EIFPORT EIF_port]"`
	where:
	XAname
	The name of the extended agent workstation.
	EIF_server
	The host name or IP address of the master domain manager.
	EIF_port
	The port that the master domain manager uses to receive the event notification.

Table 88. Miscellaneous troubleshooting items (continued)

Area	Item
`r3batch`:IDoc monitoring is not performed	**Symptom**: The events on which the event rule is based are not monitored or no event is generated during IDoc monitoring.
	Solution: The events being monitored are listed in the following file: *TWS_home*`/monconf/`*XAname*`_r3evmon.cfg` where *XAname* is the name of the SAP extended agent workstation. It is the same file that is used to monitor SAP events in general. Check that the file is updated and contains the current monitoring plan. The events corresponding to the `IDOCEventGenerated` event type are indicated by the following keyword (one for each event on the same extended agent): `!IDOC` *nnnn<Client Number>nnnn<IDoc Status List>nnnn<Direction>nnnn<Receiver Port>* *nnnn<Receiver Partner Type>nnnn<Partner Function of Receiver>* *nnnn<Partner Number of Receiver>nnnn<Sender Port>nnnn<Sender Partner Type>* *nnnn<Partner Function of Sender>nnnn<Partner Number of Sender>* *nnnn<Message Type>nnnn<IDoc Type>nnnn<Logical Message Variant>* *nnnn<Logical Message Function>nnnn<Test Flag>nnnn<Output Mode>* where: *nnnn* The length of the IDoc field. For example, `0005` indicates the value of an IDoc status list corresponding to `56,60`. < > Contains the value of the field associated with the IDoc to be monitored. For a list of the supported IDoc fields, refer to Table 51 on page 191. For each configuration file, an `r3evmon` process is started to monitor the events listed. Make sure that an `r3evmon` monitoring process is started for the involved extended agent workstation.
`r3evmon`: monitoring SAP and IDoc events increases memory consumption	**Symptom**: Memory consumption increases continuously during monitoring of IDoc and standard SAP events.
	Solution: Refer to SAP Notes 1021071 and 1109413.

Table 88. Miscellaneous troubleshooting items (continued)

Area	Item
`r3batch`: Duplicated events generated during IDoc monitoring	**Symptom**: The action defined in an event rule with `IDOCEventGenerated` event type is unexpectedly repeated.

	Solution: Reset the start date and time for the next monitoring loop. These values are stored in the following file:
	`TWS_home/methods/r3evmon_cfg/XAname_r3idocmon.cfg`
	where `XAname` is the name of the SAP extended agent workstation. Therefore you can either:
	• Stop `r3evmon`, delete the *XAname*_`r3idocmon.cfg` file and then start `r3evmon` again.
	- OR -
	• Stop `r3evmon`, set the date and time in the *XAname*_`r3idocmon.cfg` file to the values you want, and start`r3evmon` again.
	Use the following format for the start date and time:
	`start_date=YYYYMMDD`
	`start_time=HHMMSS`
	For example:
	`start_date=20080307`
	`start_time=115749`
	Check the value of the **idoc_no_history** option:
	• If it is set to OFF and no *XAname*_`r3idocmon.cfg` file exists, then all matching IDocs are retrieved, not only the current ones.
	• If it is set to ON (default value), check the date and time in the *XAname*_`r3idocmon.cfg` file.

`r3batch`: No event is generated during IDoc monitoring	**Symptom**: The expected event actions are not triggered.
	Solution: Check the value of the **idoc_no_history** option; if it is set to ON (default value), check the date and time in the *XAname*_`r3idocmon.cfg` file.

Error defining an internetwork dependency based on SAP event	**Symptom**: If you work with XBP 2.0, when you try to define an internetwork dependency based on an SAP event, the following error message is displayed:
	`*** ERROR 778 *** EEWO0778E An internal error has occurred. The program could not modify the following job:`
	`Job name:`
	`Job ID:`
	`%CJ ERROR`

	Solution: Perform the following steps: 1. Check if the BCTEST report is defined in your SAP system by invoking either one of the following transactions: **sa38** Enter BTC* and click the picklist button. In the panel that opens, click the picklist button and check if BTCTEST is shown in the list that is displayed. **se38** Enter BTC* and click the picklist button. Check if BTCTEST is shown in the list that is displayed. 2. If report BTCTEST is not found in the list, you can either: • Choose another existing report, and, in the local options file, set the placeholder_abap_step option to the name you chose. Because the report assigned to the placeholder job is run when the corresponding event is raised, ensure that you choose a dummy report. For details about the placeholder_abap_step option, see Table 35 on page 95. - OR - • Set the placeholder_abap_step option to a custom developed ABAP code of your choice.

Table 88. Miscellaneous troubleshooting items (continued)

<table>
<tr><th>Area</th><th>Item</th></tr>
<tr><td rowspan="2">r3batch: error message when scheduling SAP jobs</td><td><b>Symptom</b>: When creating an SAP job, the following message is displayed while trying to view the details of an ABAP's variant:<pre>AWSO0101E
Missing ABAP routine.
J_101_REPORT_ALL_SELECTIONS please install the latest ABAP routine for Maestro!!</pre></td></tr>
<tr><td><b>Solution</b>: This defect is caused by an error in an SAP function module. SAP describes this problem and possible solutions in the SAP Notes 0351293 and 0116354.</td></tr>
<tr><td>r3batch: modify job step error</td><td>You change print parameters with the BAPI_XBP_MODIFY_JOB_STEP function module, and subsequently, they are incorrect. As a consequence, r3batch gets error 221:<pre>MSG_CANNOT_GET_PRIARC_PARAMS: "Retrieving new print and archive parameters failed"</pre>The problem is solved by installing SAP Note 758829.</td></tr>
<tr><td>r3batch: modify job step error</td><td>The BAPI_XBP_MODIFY_JOB_STEP function module always uses the name of the logged-on user as the name for the step user. In this case, when submitting a job with the -vX options, r3batch creates a job by copying all the data from the original template, except the variant name of the first step (which is provided as the option parameter). This procedure is usually referred to as the "old copy". However, when adding a step to a new job, the XBP 2.0 interface ignores the user parameter passed by r3batch.

The problem is solved by installing SAP note 758829.</td></tr>
<tr><td rowspan="2">r3batch: does not start after installation on Windows</td><td><b>Symptom</b>: After installing or upgrading the SAP R/3 access method to version 8.5 on a Windows operating system, you try to start r3batch but nothing happens. The following message is displayed:<pre>The application failed to initialize properly.
Click on OK to terminate the application.</pre></td></tr>
<tr><td><b>Solution</b>: Ensure that you applied the SAP Note 684106 to install the required Microsoft DLLs.</td></tr>
<tr><td rowspan="2">r3batch: Tivoli Workload Scheduler environment variables are not resolved when specified in the task string for an R/3 batch job.</td><td><b>Symptom</b>: When Tivoli Workload Scheduler environment variables are used in the task string for an R/3 batch job and the job is launched, the environment variables are not resolved. The exact string used to specify the variable is used instead.</td></tr>
<tr><td><b>Solution</b>: To leverage Tivoli Workload Scheduler environment variables, you must modify the access method as follows:
1. In the <i>TWS_home</i>/methods directory, create a file named, r3batch.cmd (on Windows) or r3batch.sh (on UNIX) as required, containing the following content:<pre>@echo off
set METHODSPATH=%~dp0
call "%METHODSPATH:"=%r3batch.exe" %*</pre>2. Modify the CPU XAGENT definition from r3batch to r3batch.cmd. An example follows:<pre>CPUNAME NW1
  DESCRIPTION "r3batch"
  OS OTHER
  NODE none TCPADDR 31111
  FOR MAESTRO HOST STROMBOLI ACCESS "r3batch.cmd"
    TYPE X-AGENT
    AUTOLINK OFF
    BEHINDFIREWALL OFF
    FULLSTATUS OFF
END</pre>3. To modify the CPU access method in the Symphony™ file, run JnextPlan as follows:<pre>JnextPlan -for 0000</pre></td></tr>
</table>

Table 88. Miscellaneous troubleshooting items (continued)

Area	Item

@longlink file present in installation directory	**Symptom**: After installing IBM Tivoli Workload Scheduler for Applications on a computer with an AIX operating system where a master domain manager is already installed, a @longlink file containing the following is present in the installation directory:
	`methods/_tools/_jvm_64/lib/desktop/icons/HighContrastInverse/48x48/mimetypes/`
	`gnome-mime-text-x-java.png`

	Solution: The file can be ignored. It does not present any problems for the proper functioning of the product.

Option Editor does not start	**Symptom**: When you start the Option Editor, nothing happens.
	The following error message might be displayed:
	`The size of the extracted files to be installed are corrupted.`
	`Please attempt to download the installer again.  Make sure that the file`
	`is transfered in "binary" mode.`

	Cause and Solution: Possible causes are:
	• You upgraded Tivoli Workload Scheduler for Applications to version 8.5.
	• You added an access method to your Tivoli Workload Scheduler for Applications version 8.5 instance.
	• You upgraded the hosting Tivoli Workload Scheduler environment to version 8.5.
	To start the Option Editor, enter the following command located in *TWS_home*`/methods/opted`:
	UNIX operating systems
	`opted.sh`
	Windows operating systems
	`opted.bat`

Job throttling does not start on HP-UX	**Symptom**: You try to start the job throttler and the following error message is displayed:
	`Error occurred during initialization of VM`
	`java.lang.NoSuchMethodError: java.lang.Thread.start()V Abort`

	Cause and Solution: Java version 1.5 does not start because there are symbolic links of Java version 1.4 libraries used by third party products. For example, you might have `/usr/lib/pa20_64/libjava.sl` linked to `/opt/java1.4/jre/lib/PA_RISC2.0W/libjava.sl`
	Before starting the job throttling again, change the PATH and SHLIB_PATH environment variables as follows:
	`PATH=`*TWS_home*`/methods/_tools/_jvm_64/lib/PA_RISC2.0W:$PATH`
	`export PATH`
	`SHLIB_PATH=`*TWS_home*`/methods/_tools/_jvm_64/lib/PA_RISC2.0W:$SHLIB_PATH`
	`export SHLIB_PATH`
	To apply this change definitively, edit the `jobthrottling.sh` file by adding the environment settings after the following line:
	`#!/bin/sh`

Table 88. Miscellaneous troubleshooting items (continued)

Area	Item

Job throttling does not start	**Symptom**: When you start the job throttling feature, nothing happens and the following error message is displayed:
	`EEWOJTR0207E Error, another job throttler instance is already running`
	`against the same SAP system. Foreign job throttler registration is:`
	`Client ID="`*`clientID`*`", Name="TWS4APPS_JOBTHROTTLER",Host="`*`hostname`*`",`
	`UID "`*`UniqueID`*`"`

	Cause and Solution: Possible causes are: • You are running job interception collector jobs, but the job interception and job throttling features cannot run at the same time. Choose which feature to start. For detailed information, refer to “Job interception and parent-child features” on page 150 and “Job throttling feature” on page 173. • Another job throttler instance is running against the same SAP system. You can start only one job throttler instance. • A previous job throttler instance created an exclusive lock object on the SAP system that could have become permanent. To verify it, use transaction `sm12` and query for the lock object named `TWS4APPS_JOBTHROTTLER`. If the lock object exists, and you are not running any job throttler or job interception instance, remove the lock manually and restart the job throttler.
Job throttling does not start	**Symptom**: When you start the job throttling feature, nothing happens and the following error message is displayed: `EEWOJT0209E Error, the password format is not valid.`
	Cause and Solution: Your password is encrypted in old format. To encrypt the password with the correct encryption version, use the **enigma** or **pwdcrypt** programs, or the Option Editor. For details about how to encrypt the password, see “Encrypting SAP R/3 user passwords” on page 102.
Job throttling does not stop	**Symptom**: When you stop the job throttling feature, nothing happens.
	Cause and Solution: You are connected as a *TWSUser* who does not have write permission on the *`XAname`*`_jobthrottling_cmd.txt` file. To solve this problem, delete the *`XAname`*`_jobthrottling_cmd.txt` file and enter the command again. For detailed information about stopping the job throttler, refer to “Step 5. Starting and stopping the job throttling feature” on page 176.
Job throttling: alerts for MTEs are not generated according to the threshold values set	**Symptom**: Alerts for the MTEs created by the job throttler are generated without respecting the threshold values that are set.
	Cause and Solution: You started a new job throttler instance, which, being enabled to send data to CCMS, created the related MTEs. When you include the MTEs in your monitoring set, the threshold values are automatically set according to the existing MTE class. Nevertheless, alerts are generated without respecting these values. To solve this problem, edit the MTE properties and save them again, even if you do not change anything.
Job throttling: saving MTE properties generates an informational message	**Symptom**: When you edit and save the properties of MTEs generated by the job throttler, the following informational message is displayed: `Message does not exist.`
	Cause and Solution: In the pop-up window that displays the message, click **Continue** and close the Properties window. Your settings are saved.

Job throttling: error message displayed when creating trace file on HP operating systems	**Symptom**: While the job throttler is stopping, there are intercepted jobs to release on exit. The following error message is displayed:
	`CJL0006E Handler jobthrottling.trace.handlers.file`
	`is unable to write a log event.`

	Cause and Solution: The message does not report any real error, and can be ignored.

Table 88. Miscellaneous troubleshooting items (continued)

Area	Item
The system cannot intercept jobs	**Symptom**: Although the job interception feature is active on the SAP system, the intercepted jobs are kept in scheduled state.
	Cause and Solution: The job throttler feature or the Java Virtual Machine used by the job throttler might still be active. On each extended agent where the job throttler was started at least once, ensure that: 1. You stopped the feature. For details, see "Step 5. Starting and stopping the job throttling feature" on page 176. 2. The Java Virtual Machine used by the job throttler was stopped by the process. To search for Java processes, use: **On Windows** The Process Explorer **On UNIX** The command `ps -ef | grep throttling` If a Java Virtual Machine instance related to the job throttler is found, kill it.
`access method executables: r3batch, r3event, psagent:` permission denied messages in the job log.	**Symptom**: The job log reports multiple "Permission denied" messages.

	Cause and Solution: The root cause might be that the access method executable, for example, r3batch, is submitted by the `root` user and not the `twsuser`. This creates directories and files with the wrong ownership and file permissions. Verify the ownership of the following directories and files if you are running the product on UNIX platforms. Ensure that the `twsuser` is the owner of the files and that the user has both read and write permissions on the files, and execute permission on the directories.
	`<TWShome>/methods/traces`
	`<TWShome>/methods/traces/*.log`

`psagent`: misleading message displayed if the local options file has no right permissions	**Symptom**: The job log shows the following message:
	`EEW00439E You did not specify the required options either in the global`
	`or in the local options file.`
	but all the mandatory options were correctly set in the options file.

	Solution: Check that the options file has `read` and `write` permissions available to the user who is trying to launch the job.
No messages written in the job log	**Symptom**: Tivoli Workload Scheduler for Applications does not write any messages in the job log if the file system for tracing is full or the ljuser does not have the correct permission to write in the trace directory.

The submission of a PeopleSoft job fails	**Symptom**: The submission of a PeopleSoft job fails and the Tivoli Workload Scheduler job log contains a Java exception similar to the following:
	`Exception in thread "3194" java.lang.ExceptionInInitializerError`
	`        at bea.jolt.JoltSessionAttributes.<clinit>(JoltSessionAttributes.java:183)`
	`        at psft.pt8.net.JoltSessionPool.createConnection(JoltSessionPool.java:363)`
	`        at psft.pt8.net.JoltSessionPool.getJoltSession(JoltSessionPool.java:220)`

	Cause and Solution: The `psjoa.jar` path contains special characters. Define a path without special characters.

The submission of an Oracle job fails	**Symptom**: The submission of an Oracle job fails and the Tivoli Workload Scheduler job log shows the following information:
	`EEWP0017 Child MCMLJ exited normally.`
	`Exit code: 1.EEWP0027 Error - Launch job failed`

	Solution: Submitting an Oracle job might fail because there is a connection problem to the Oracle database. Verify that your Oracle naming methods are set correctly. For details about how to configure naming methods, refer to the *Oracle Net Services Administrator's Guide*.

Table 88. Miscellaneous troubleshooting items (continued)

Area	Item
mvsjes: RACF authorization problem on z/OS version 1.7	**Symptom**: An S047 abend is returned if the EEWSERVE started task does not have an associated RACF owner ID.
	Solution: In the RACF database, associate an authorized RACF ID with the EEWSERVE started task as specified in "Setting RACF authorizations on z/OS" on page 213.

Troubleshooting the SAP R/3 connection

If you are unable to submit SAP R/3 jobs using Tivoli Workload Scheduler after the R/3 configuration, perform the following tests:

- Ensure that you can ping the SAP R/3 system from the Tivoli Workload Scheduler system. This shows basic network connectivity.
- Note that using the SAP R/3 routers to access the R/3 system could exceed the size of internal buffers of the RFC library used to store the hostname of the SAP R/3 system. When this occurs, the hostname gets truncated, causing the connection to the R/3 system to fail. To work around this problem, do not fully qualify the name of the SAP R/3 routers or alternatively use the IP addresses.
- Run the following **telnet** command to verify connectivity:

  ```
  telnet systemname 33xx
  ```

  where *systemname* is the system name or IP address of the SAP R/3 server and *xx* is the SAP R/3 instance.

  If the command fails to complete, this means that communication between r3batch and the SAP R/3 application server is down.
- Log on to the SAP R/3 system as an administrator and verify that the Tivoli Workload Scheduler RFC user (created in the "Creating the IBM Tivoli Workload Scheduler RFC user" on page 80) exists.
- If the SAP R/3 gateway truncates the connection string, replace the host name with the IP address.
- If r3batch runs on an AIX system that does not use U.S. English, make sure that the U.S. Language Environment® is installed on both the IBM Tivoli Workload Scheduler workstation and the SAP R/3 database workstation. Otherwise the error BAD TEXTENV (or a similar error message) might appear in the dev_rfc trace file and connections to SAP R/3 fail.

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:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

If you are viewing this information in softcopy form, the photographs and color illustrations might not appear.

Notices and information

The IBM license agreement and any applicable notices on the web download page for this product refers You to this file for details concerning terms and conditions applicable to software code identified as excluded components in the License Information document and included in IBM Tivoli Workload Scheduler for Applications 8.4.0 (the "Program").

Notwithstanding the terms and conditions of any other agreement you may have with IBM or any of its related or affiliated entities (collectively "IBM"), the third party software code identified below are "Excluded Components" and are subject to the terms and conditions of the License Information document accompanying the Program and not the license terms that may be contained in the notices below. The notices are provided for informational purposes.

The Program includes the following Excluded Components:
- libmsg
- Jakarta ORO
- ISMP Installer
- HSQLDB
- Quick
- Infozip

Libmsg

For the code entitled Libmsg

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that Alfalfa's name not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

ALPHALPHA DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL ALPHALPHA BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Apache Jakarta ORO

For the code entitled Jakarta ORO

The Apache Software License, Version 1.1

Copyright (c) 2000-2002 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Apache" and "Apache Software Foundation", "Jakarta-Oro" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
5. Products derived from this software may not be called "Apache" or "Jakarta-Oro", nor may "Apache" or "Jakarta-Oro" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see http://www.apache.org/.

Portions of this software are based upon software originally written by Daniel F. Savarese. We appreciate his contributions.

ISMP Installer (InstallShield 10.50x)

For the code entitled ISMP Installer (InstallShield 10.50x)

The Program includes the following Excluded Components:
- Quick V1.0.1
- HSQLDB V1.7.1
- InfoZip Unzip stub file V5.40, V5.41,V5.42 & V5.5

JXML CODE

For the code entitled Quick

JXML CODE. The Program is accompanied by the following JXML software:
- Quick V1.0.1

IBM is required to provide you, as the recipient of such software, with a copy of the following license from JXML:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

All product materials mentioning features or use of this software must display the following acknowledgement:

- This product includes software developed by JXML, Inc. and its contributors: http://www.jxml.com/mdsax/contributers.html

Neither name of JXML nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY JXML, INC. AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JXML OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

InfoZip CODE

For the code entitled InfoZip

InfoZip CODE. The Program is accompanied by the following InfoZip software:

- One or more of: InfoZip Unzipsfx stub file V5.40, V5.41, V5.42 & V5.5

IBM is required to provide you, as the recipient of such software, with a copy of the following license from InfoZip:

- This is version 2000-Apr-09 of the Info-ZIP copyright and license.

  The definitive version of this document should be available at ftp://ftp.info-zip.org/pub/infozip/license.html indefinitely.

Copyright (c) 1990-2000 Info-ZIP. All rights reserved.

For the purposes of this copyright and license, "Info-ZIP" is defined as the following set of individuals:

- Mark Adler, John Bush, Karl Davis, Harald Denker, Jean-Michel Dubois, Jean-loup Gailly, Hunter Goatley, Ian Gorman, Chris Herborth, Dirk Haase, Greg Hartwig, Robert Heath, Jonathan Hudson, Paul Kienitz, David Kirschbaum, Johnny Lee, Onno van der Linden, Igor Mandrichenko, Steve P. Miller, Sergio Monesi, Keith Owens, George Petrov, Greg Roelofs, Kai Uwe Rommel, Steve Salisbury, Dave Smith, Christian Spieler, Antoine Verheijen, Paul von Behren, Rich Wales, Mike White

This software is provided "as is," without warranty of any kind, express or implied. In no event shall Info-ZIP or its contributors be held liable for any direct, indirect, incidental, special or consequential damages arising out of the use of or inability to use this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. Redistributions of source code must retain the above copyright notice, definition, disclaimer, and this list of conditions.
2. Redistributions in binary form must reproduce the above copyright notice, definition, disclaimer, and this list of conditions in documentation and/or other materials provided with the distribution.
3. Altered versions--including, but not limited to, ports to new operating systems, existing ports with new graphical interfaces, and dynamic, shared, or static library versions--must be plainly marked as such and must not be misrepresented as being the original source. Such altered versions also must not be misrepresented as being Info-ZIP releases--including, but not limited to, labeling of the altered versions with the names "Info-ZIP" (or any variation thereof, including, but not limited to, different capitalizations), "Pocket UnZip," "WiZ" or "MacZip" without the explicit permission of Info-ZIP. Such altered versions are further prohibited from misrepresentative use of the Zip-Bugs or Info-ZIP e-mail addresses or of the Info-ZIP URL(s).
4. Info-ZIP retains the right to use the names "Info-ZIP," "Zip," "UnZip," "WiZ," "Pocket UnZip," "Pocket Zip," and "MacZip" for its own source and binary releases.

HSQL Code

For the code entitled HSQLDB

HSQL CODE. The Program is accompanied by the following HSQL Development Group software:

- HSQLDB V1.7.1

IBM is required to provide you, as the recipient of such software, with a copy of the following license from the HSQL Development Group:

Copyright (c) 2001-2002, The HSQL Development Group All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the HSQL Development Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

HP-UX Runtime Environment, for the Java 2 Platform

TERMS FOR SEPARATELY LICENSED CODE

This Program includes HP Runtime Environment for J2SE HP-UX 11i platform software as a third party component, which is licensed to you under the terms of the following HP-UX license agreement and not those of this Agreement

HP-UX Runtime Environment license text

HP-UX Runtime Environment, for the Java 2 Platform

ATTENTION: USE OF THE SOFTWARE IS SUBJECT TO THE HP SOFTWARE LICENSE TERMS AND SUPPLEMENTAL RESTRICTIONS SET FORTH BELOW, THIRD PARTY SOFTWARE LICENSE TERMS FOUND IN THE THIRDPARTYLICENSEREADME.TXT FILE AND THE WARRANTY DISCLAIMER ATTACHED. IF YOU DO NOT ACCEPT THESE TERMS FULLY, YOU MAY NOT INSTALL OR OTHERWISE USE THE SOFTWARE. NOTWITHSTANDING ANYTHING TO THE CONTRARY IN THIS NOTICE, INSTALLING OR OTHERWISE USING THE SOFTWARE INDICATES YOUR ACCEPTANCE OF THESE LICENSE TERMS.

HP SOFTWARE LICENSE TERMS

The following terms govern your use of the Software unless you have a separate written agreement with HP. HP has the right to change these terms and conditions at any time, with or without notice.

License Grant

HP grants you a license to Use one copy of the Software. "Use" means storing, loading, installing, executing or displaying the Software. You may not modify the Software or disable any licensing or control features of the Software. If the Software is licensed for "concurrent use", you may not allow more than the maximum number of authorized users to Use the Software concurrently.

Ownership

The Software is owned and copyrighted by HP or its third party suppliers. Your license confers no title or ownership in the Software and is not a sale of any rights in the Software. HP's third party suppliers may protect their rights in the event of any violation of these License Terms.

Third Party Code

Some third-party code embedded or bundled with the Software is licensed to you under different terms and conditions as set forth in the THIRDPARTYLICENSEREADME.txt file. In addition to any terms and conditions of any third party license identified in the THIRDPARTYLICENSEREADME.txt file, the disclaimer of warranty and limitation of liability provisions in this license shall apply to all code distributed as part of or bundled with the Software.

Source Code

Software may contain source code that, unless expressly licensed for other purposes, is provided solely for reference purposes pursuant to the terms of this license. Source code may not be redistributed unless expressly provided for in these License Terms.

Copies and Adaptations

You may only make copies or adaptations of the Software for archival purposes or when copying or adaptation is an essential step in the authorized Use of the Software. You must reproduce all copyright notices in the original Software on all copies or adaptations. You may not copy the Software onto any bulletin board or similar system.

No Disassembly or Decryption

You may not disassemble or decompile the Software unless HP's prior written consent is obtained. In some jurisdictions, HP's consent may not be required for disassembly or decompilation. Upon request, you will provide HP with reasonably detailed information regarding any disassembly or decompilation. You may not decrypt the Software unless decryption is a necessary part of the operation of the Software.

Transfer

Your license will automatically terminate upon any transfer of the Software. Upon transfer, you must deliver the Software, including any copies and related documentation, to the transferee. The transferee must accept these License Terms as a condition to the transfer.

Termination

HP may terminate your license upon notice for failure to comply with any of these License Terms. Upon termination, you must immediately destroy the Software, together with all copies, adaptations and merged portions in any form.

Export Requirements

You may not export or re-export the Software or any copy or adaptation in violation of any applicable laws or regulations.

This software or any copy or adaptation may not be exported, reexported or transferred to or within countries under U.S. economic embargo including the following countries: Afghanistan (Taliban-controlled areas), Cuba, Iran, Iraq, Libya, North Korea, Serbia, Sudan and Syria. This list is subject to change.

This software or any copy or adaptation may not be exported, reexported or transferred to persons or entities listed on the U.S. Department of Commerce Denied Parties List or on any U.S. Treasury Department Designated Nationals exclusion list, or to any party directly or indirectly involved in the development or production of nuclear, chemical, biological weapons or related missile technology programs as specified in the U.S. Export Administration Regulations (15 CFR 730).

U.S. Government Contracts

If the Software is licensed for use in the performance of a U.S. government prime contract or subcontract, you agree that, consistent with FAR 12.211 and 12.212, commercial computer Software, computer Software documentation and technical data for commercial items are licensed under HP's standard commercial license.

SUPPLEMENTAL RESTRICTIONS

You acknowledge the Software is not designed or intended for use in on-line control of aircraft, air traffic, aircraft navigation, or aircraft communications; or in the design, construction, operation or maintenance of any nuclear facility. HP disclaims any express or implied warranty of fitness for such uses.

ADDITIONAL SUPPLEMENTAL RESTRICTIONS FOR HP-UX RUNTIME ENVIRONMENT, FOR THE JAVA(TM) 2 PLATFORM

- * License to Distribute HP-UX Runtime Environment, for the Java(tm) 2 Platform. You are granted a royalty-free right to reproduce and distribute the HP-UX Runtime Environment, for Java provided that you distribute the HP-UX Runtime Environment, for the Java 2 Platform complete and unmodified, only as a part of, and for the sole purpose of running your Java compatible applet or application ("Program") into which the HP-UX Runtime Environment, for the Java 2 Platform is incorporated.
- * Java Platform Interface. Licensee may not modify the Java Platform Interface ("JPI", identified as classes contained within the "java" package or any subpackages of the "java" package), by creating additional classes within the JPI or otherwise causing the addition to or modification of the classes in the JPI. In the event that Licensee creates any Java-related API and distributes such API to others for applet or application development, Licensee must promptly publish broadly, an accurate specification for such API for free use by all developers of Java-based software.
- * You may make the HP-UX Runtime Environment, for the Java 2 Platform accessible to application programs developed by you provided that the programs allow such access only through the Invocation Interface specified and provided that you shall not expose or document other interfaces that permit access to such HP-UX Runtime Environment, for the Java 2 Platform. You shall not be restricted hereunder from exposing or documenting interfaces to software components that use or access the HP-UX Runtime Environment, for the Java 2 Platform.

HP WARRANTY STATEMENT

DURATION OF LIMITED WARRANTY: 90 DAYS

HP warrants to you, the end customer, that HP hardware, accessories, and supplies will be free from defects in materials and workmanship after the date of purchase for the period specified above. If HP receives notice of such defects during the warranty period, HP will, at its option, either repair or replace products which prove to be defective. Replacement products may be either new or equivalent in performance to new.

HP warrants to you that HP Software will not fail to execute its programming instructions after the date of purchase, for the period specified above, due to defects in materials and workmanship when properly installed and used. If HP receives notice of such defects during the warranty period, HP will replace Software which does not execute its programming instructions due to such defects.

HP does not warrant that the operation of HP products will be uninterrupted or error free. If HP is unable, within a reasonable time, to repair or replace any product to a condition warranted, you will be entitled to a refund of the purchase price upon prompt return of the product. Alternatively, in the case of HP Software, you will be entitled to a refund of the purchase price upon prompt delivery to HP of written notice from you confirming destruction of the HP Software, together with all copies, adaptations, and merged portions in any form.

HP products may contain remanufactured parts equivalent to new in performance or may have been subject to incidental use.

Warranty does not apply to defects resulting from: (a) improper or inadequate maintenance or calibration; (b) software, interfacing, parts or supplies not supplied by HP, (c) unauthorized modification or misuse; (d) operation outside of the published environmental specifications for the product, (e) improper site preparation or maintenance, or (f) the presence of code from HP suppliers embedded in or bundled with any HP product.

TO THE EXTENT ALLOWED BY LOCAL LAW, THE ABOVE WARRANTIES ARE EXCLUSIVE AND NO OTHER WARRANTY OR CONDITION, WHETHER WRITTEN OR ORAL, IS EXPRESSED OR IMPLIED AND HP SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, SATISFACTORY QUALITY, AND FITNESS FOR A PARTICULAR PURPOSE. Some countries, states, or provinces do not allow limitations on the duration of an implied warranty, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights and you might also have other rights that vary from country to country, state to state, or province to province.

TO THE EXTENT ALLOWED BY LOCAL LAW, THE REMEDIES IN THIS WARRANTY STATEMENT ARE YOUR SOLE AND EXCLUSIVE REMEDIES. EXCEPT AS INDICATED ABOVE, IN NO EVENT WILL HP OR ITS SUPPLIERS BE LIABLE FOR LOSS OF DATA OR FOR DIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL (INCLUDING LOST PROFIT OR DATA), OR OTHER DAMAGE, WHETHER BASED IN CONTRACT, TORT, OR OTHERWISE. Some countries, states, or provinces do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation may not apply to you.

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.

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 TM), these symbols indicate US 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

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

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

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

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

SAP and SAP NetWeaver are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

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

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Index

Printed in USA

SC32-1278-07

Spine information:

IBM Tivoli Workload Scheduler for Applications **Version 8.6** **User’s Guide**