

IBM Tivoli Workload Scheduler for Applications



User's Guide

Version 8.2

IBM Tivoli Workload Scheduler for Applications



User's Guide

Version 8.2

Note

Before using this information and the product it supports, read the information in "Notices" on page 189.

First Edition (June 2003)

This edition applies to version 8, release 2, modification 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.

© Copyright International Business Machines Corporation 2000, 2003. All rights reserved.

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

Contents

Preface	vii
Who Should Read This Manual	vii
What This Manual Contains.	vii
Publications	ix
Publications for IBM Tivoli Workload Scheduler	ix
Publications for IBM Tivoli Workload Scheduler for z/OS.	x
Softcopy Collection Kit for IBM Tivoli Workload Scheduler for z/OS	xi
Related publications	xi
Accessing Publications Online	xi
Ordering Publications	xii
Providing Feedback about Publications	xii
Accessibility	xii
Contacting IBM Software Support.	xii
Conventions Used in This Manual	xiii
Typeface conventions	xiii
Operating system-dependent variables and paths	xiii

Part 1. Common Installation and Configuration Tasks 1

Chapter 1. Product Overview. 3

Chapter 2. Installing IBM Tivoli Workload Scheduler for Applications 5

How to Install	6
Using the Installation Wizard	7
Using Silent Installation	17
Using Tivoli Configuration Manager	20
Uninstalling the Product	23
Removing an Installation via ISMP	23
Removing an Installation via Tivoli Configuration Manager	24
Completing the Removal of the Access Method for Oracle e-Business Suite	24

Chapter 3. Setting Options with the Option Editor 25

Creating a Local Options File	27
Option Value Inheritance	28

Chapter 4. Defining Extended Agent Workstations 29

Creating a Workstation with the Job Scheduling Console.	29
Creating a Workstation from the Command Line	31
Defining Workstations for End-to-end Scheduling	32
Creating the CPUREC Statement	32
Defining the Workstation with ISPF	33

Chapter 5. Defining Extended Agent Jobs. 35

Defining Jobs with the Job Scheduling Console	35
Defining Jobs with the Tivoli Workload Scheduler Command Line	38
Defining Jobs for End-to-end Scheduling	39

Part 2. Oracle e-Business Suite Access Method 41

Chapter 6. Introduction. 43

Features	43
Software Requirements	43
Post-installation Tasks	43
Interface Software	45

Chapter 7. Job Definition Information 47

Defining Jobs in Oracle Applications	47
Running a Single Job Concurrently	47
Defining Oracle Applications Jobs in Tivoli Workload Scheduler	47
Task String Parameters for Oracle Applications Jobs	47
Tivoli Workload Scheduler and Oracle Applications Job States	49

Chapter 8. MCMAGENT Options File Reference 51

The Options File.	51
Entry Syntax	51
Example	52
Encrypting Oracle Applications User Passwords	52

Part 3. PeopleSoft Access Method 53

Chapter 9. Introduction. 55

Features	55
Functional Overview	55
PeopleSoft Process Scheduler Jobs	55
PeopleSoft Job Tracking in Tivoli Workload Scheduler	56
Security	56
Notes on PeopleSoft Version 8.x Support	56

Chapter 10. Implementation Planning 59

PeopleSoft Version 7.5 or Earlier	59
Host Requirements	59
Software Requirements	59
Windows User Setup	59
Verifying the PeopleSoft Environment	60
Creating a Batch Processing ID in PeopleSoft	60
Installed Files.	60

PeopleSoft Version 8.x	60
Host Requirements	60
Software Requirements	60
Creating a Batch Processing ID in PeopleSoft	61
Installed Files.	61

Chapter 11. Post-installation Tasks. 63

User Setup for Windows	63
Creating the Options File	63
Manually Editing Options Files.	66
Connecting to More than One PeopleSoft Database	67
Setting Up the PeopleSoft Environment for Version 7.5 or earlier	67
Customizing the Process Server.	67
Setting Up the Client	69
Running the Data Mover Script.	70
Setting Up the PeopleSoft Environment for Version 8.1x	72
Customizing the Application Server	72
Setting Up the Client	72
Setting Up the PeopleSoft Environment for Version 8.4x	73

Chapter 12. Defining PeopleSoft Jobs 75

Creating Peoplesoft Job Definitions in Tivoli Workload Scheduler	75
Task String Parameters for PeopleSoft Jobs	75
Setting up Opens Dependencies 7.x Only	76
Example Opens Dependencies	77
IBM Tivoli Workload Scheduler and PeopleSoft Job Status Mapping	78

Part 4. R/3 Access Method 81

Chapter 13. Introduction 83

Features	83
Scheduling Process for the R/3 Extended Agent	83
XBP 2.0 Interface Support.	84

Chapter 14. Installation. 85

Software Requirements	85
Installation Overview	85
Installing on Tier2 Platforms.	86
Encrypting R/3 User Passwords	87

Chapter 15. R/3 Configuration 89

Overview	89
Creating the IBM Tivoli Workload Scheduler RFC User.	89
Creating the Authorization Profile for the Tivoli Workload Scheduler User.	89
Using Transaction SU02 and Manually Creating the Profile	89
Using Transaction PFCG (Profile Generator)	90
Copy the Correction And Transport Files	92
Import ABAP/4 Modules into R/3	92
Installing the r3batch ABAP Modules on the R/3 System	93

Troubleshooting	93
Changing the Tivoli Workload Scheduler RFC User ID Password	94
Migrating from Previous Versions	95
Using XBP 2.0 Support	95
Creating The Authorization Profile for the Tivoli Workload Scheduler User.	96
Print Parameter and Job Class Issues	96

Chapter 16. Defining R/3 Extended Agent Jobs 99

Defining a Job in R/3	99
Defining a Job in Tivoli Workload Scheduler	99
Using the SAP-Specific Windows of the Job Scheduling Console	100
Task String Parameters for R/3 Jobs.	107
Tivoli Workload Scheduler and R/3 Job States	109

Chapter 17. Re-running Jobs 111

Old Copy and New Copy of a Re-running Job	111
---	-----

Chapter 18. Business Information Warehouse 113

Business Information Warehouse Support - InfoPackages	113
User Authorizations	113
Business Warehouse Components.	113
Defining Extended Agent Jobs for BIW	113
R/3 Schedule Options of an InfoPackage	114

Chapter 19. Connecting to the R/3 System 115

Connections to a Specific Application Server	115
Logon Groups	116

Chapter 20. R/3 Access Method Options Files 117

Configuration Options	117
Global Options	117
Local Options	118
General Options	119
Configuration Options Usage	121
Configuration Files Migration	122

Chapter 21. BDC Wait 125

Using BDC Session Names	125
Using BDC Queue IDs with Report BDCRECXX	126
Using BDC queue IDs with FM	
BDC_OPEN_GROUP	127

Chapter 22. Job Interception and Parent-child Features 129

Job Interception	129
Implementing Job Interception	129
The Parent-child Feature.	132

Chapter 23. National Language Support 133

National Language Support Options	133
R/3 Supported Codepages	133
Supported Configurations and Limitations	134

Part 5. z/OS Access Method 135

Chapter 24. Introduction 137

Features	137
Software Requirements	137
Installation Overview	137
Installing on Tier2 Platforms	138
Installing on z/OS	138
Setting Authorizations on z/OS	140
Additional Notes	140
z/OS Gateway Version	145

Chapter 25. Set Up and Operation . . . 147

Method Options Files	147
Method Options Files on Tier2 Platforms	147
Defining Jobs in z/OS	148
Defining z/OS Jobs in Tivoli Workload Scheduler	148
Task Definition Syntax for z/OS Jobs Scheduled with Tivoli Workload Scheduler	149
Task Definition Syntax for Other z/OS Jobs	150
File (OPENS) Dependencies Syntax on z/OS Files	152

Chapter 26. Reference Information 153

Technical Overview	153
Operation with JES	153
Operation with CA-7	154
Other CA-7 Tips	155
Operation with OPC or Tivoli Workload Scheduler for z/OS	155
Checking for Files on z/OS	156
Timing Considerations	157
Diagnostic Information	157
Gateway Messages	158

Part 6. Common Serviceability . . . 161

Chapter 27. Extended Agent Reference 163

What are Extended Agents?	163
-------------------------------------	-----

Workstation Definition	163
Access Method Interface	163
Method Command Line Syntax	163
Method Response Messages	165
Method Options File	166
Method Execution	167
Launch Job (LJ) Task	167
Manage Job (MJ) Task	168
Check File (CF) Task	168
Get Status (GS) Task	169
The cpuserinfo Command	169
Troubleshooting	169
Job Standard List Error Messages	170
Method Not Executable	170
Console Manager Messages	170
Composer and Compiler Messages	170
Jobman Messages	170

Chapter 28. The Return Code Mapping Library 173

The Return Code Mapping File	173
Return Code Mapping for Psagent and MCMAGENT	174
Return Code Mapping File Names	177

Chapter 29. Configuring the Tracing Utility 179

Chapter 30. Troubleshooting 181

Troubleshooting the Installation of the PeopleSoft Access Method	181
--	-----

Chapter 31. Installed Files 183

Part 7. Appendixes 187

Notices	189
Trademarks	191

Glossary 193

Index 197

Preface

The *IBM Tivoli Workload Scheduler for Applications User's Guide* provides information on 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
- R/3
- z/OS™

This guide includes and extends the contents of the following manuals:

- *IBM Tivoli Workload Scheduler for Applications Version 8.1.1 User's Guide for Oracle e-Business Suite*
- *IBM Tivoli Workload Scheduler for Applications Version 8.1.1 User's Guide for PeopleSoft*
- *IBM Tivoli Workload Scheduler for Applications Version 8.1.1 User's Guide for R/3*
- *IBM Tivoli Workload Scheduler for Applications Version 8.1.1 User's Guide for z/OS*

Who Should Read This Manual

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

- IBM Tivoli Workload Scheduler
- Tivoli Job Scheduling Console
- The specific application environment

What This Manual Contains

This manual contains the following sections:

Part 1. Common Installation and Configuration Tasks

Part 1 contains the following chapters:

- Chapter 1, "Product Overview", on page 3
describes the basic functioning of an extended agent.
- Chapter 2, "Installing IBM Tivoli Workload Scheduler for Applications", on page 5
describes the various ways of installing and uninstalling the product, together with supported platforms and prerequisites.
- Chapter 3, "Setting Options with the Option Editor", on page 25
describes how to use the Option Editor.
- Chapter 4, "Defining Extended Agent Workstations", on page 29
describes how to define extended agent workstations for all the access methods.
- Chapter 5, "Defining Extended Agent Jobs", on page 35
describes how to define extended agent jobs for all the access methods.

Part 2. Oracle e-Business Suite Access Method

Part 2 contains the following chapters:

- Chapter 6, “Introduction”, on page 43
describes features, software requirements, post-installation tasks and interface software specifically for the Oracle e-Business Suite access method.
- Chapter 7, “Job Definition Information”, on page 47
provides information for defining Tivoli Workload Scheduler jobs specifically for this access method.
- Chapter 8, “MCMAGENT Options File Reference”, on page 51
describes the options used by this access method.

Part 3. PeopleSoft Access Method

Part 3 contains the following chapters:

- Chapter 9, “Introduction”, on page 55
describes features, functions, and other general information specific to the PeopleSoft access method.
- Chapter 10, “Implementation Planning”, on page 59
describes host and software requirements and other information relevant to different PeopleSoft versions supported by this access method.
- Chapter 11, “Post-installation Tasks”, on page 63
describes post-installation tasks for this access method.
- Chapter 12, “Defining PeopleSoft Jobs”, on page 75
provides information for defining Tivoli Workload Scheduler jobs specifically for this access method.

Part 4. R/3 Access Method

Part 4 contains the following chapters:

- Chapter 13, “Introduction”, on page 83
describes the features of the R/3 access method.
- Chapter 14, “Installation”, on page 85
provides installation information specific to this access method and describes how to install on Tier 2 platforms.
- Chapter 15, “R/3 Configuration”, on page 89
describes how to configure this access method for use with R/3.
- Chapter 16, “Defining R/3 Extended Agent Jobs”, on page 99
provides information for defining Tivoli Workload Scheduler jobs specifically for the R/3 access method.
- Chapter 17, “Re-running Jobs”, on page 111
describes the mechanism for re-running jobs.
- Chapter 18, “Business Information Warehouse”, on page 113
provides information on how to use the access method with the R/3 Business Information Warehouse.
- Chapter 19, “Connecting to the R/3 System”, on page 115
describes R/3 connections and logon groups.
- Chapter 20, “R/3 Access Method Options Files”, on page 117

- describes the options used by this access method.
- Chapter 21, “BDC Wait”, on page 125
provides details on BDC Wait.
- Chapter 22, “Job Interception and Parent-child Features”, on page 129
describes the Job interception and Parent-child features of XBP 2.0.
- Chapter 23, “National Language Support”, on page 133
describes national language support for this access method.

Part 5. z/OS Access Method

Part 5 contains the following chapters:

- Chapter 24, “Introduction”, on page 137
lists the features of the z/OS access method; describes how to install the distributed part on tier 2 platforms, and how to install the gateway component on z/OS.
- Chapter 25, “Set Up and Operation”, on page 147
describes the options of this access method and provides method-specific information on defining jobs.
- Chapter 26, “Reference Information”, on page 153
provides reference information for operating with the supported job schedulers and lists gateway messages.

Part 6. Common Serviceability

Part 6 contains the following chapters:

- Chapter 27, “Extended Agent Reference”, on page 163
provides reference information for developing a Tivoli Workload Scheduler access method.
- Chapter 28, “The Return Code Mapping Library”, on page 173
describes how to use the return code mapping utility.
- Chapter 29, “Configuring the Tracing Utility”, on page 179
describes how to configure the tracing utility.
- Chapter 30, “Troubleshooting”, on page 181
Provides some general troubleshooting information.
- Chapter 31, “Installed Files”, on page 183
list installed files according to platform and feature.

Publications

This section lists publications in the IBM Tivoli Workload Scheduler library. It also describes how to access Tivoli publications online, how to order Tivoli publications, and how to make comments on Tivoli publications.

Publications for IBM Tivoli Workload Scheduler

This section lists publications in the IBM Tivoli Workload Scheduler library and related documents. It also describes how to access Tivoli publications online and how to order Tivoli publications.

- *IBM Tivoli Workload Scheduler Planning and Installation Guide*, SC32-1273

Describes how to plan a Tivoli Workload Scheduler network and how to install and customize Tivoli Workload Scheduler and its associated software.

- *IBM Tivoli Workload Scheduler Reference Guide*, SC32-1274
Explains the Tivoli Workload Scheduler command line, understanding how extended and network agents work, and integrating Tivoli Workload Scheduler with NetView® and with Tivoli Business Systems Manager.
- *IBM Tivoli Workload Scheduler Troubleshooting and Error Messages*, SC32-1275
Interprets Tivoli Workload Scheduler error messages, and provides sources of information that will help you in solving problems with Tivoli Workload Scheduler.
- *IBM Tivoli Workload Scheduler Job Scheduling Console Release Notes*, SC32-1258
Provides information about working with Tivoli Workload Scheduler, regardless of platform, from a common GUI.
- *IBM Tivoli Workload Scheduler Job Scheduling Console User's Guide*, SC32-1257
Provides information about working with Tivoli Workload Scheduler, regardless of platform, from a common GUI.
- *IBM Tivoli Workload Scheduler for Applications Release Notes*, SC32-1279
Provides last-minute information about the Tivoli Workload Scheduler extended agents.
- *IBM Tivoli Workload Scheduler AS/400® Limited Fault-tolerant Agent User's Guide*, SC32-1280
Describes installing, configuring, and using Tivoli Workload Scheduler fault-tolerant agents on AS/400.
- *IBM Tivoli Workload Scheduler Plus Module User's Guide*, SC32-1276
Explains setting up and using the Tivoli Workload Scheduler Plus module.
- *IBM Tivoli Workload Scheduler Release Notes*, SC32-1277
Explains late-breaking information about Tivoli Workload Scheduler.

Publications for IBM Tivoli Workload Scheduler for z/OS

The following documents are available in the IBM Tivoli Workload Scheduler for z/OS library:

- *IBM Tivoli Workload Scheduler for z/OS Customization and Tuning*, SC32-1265
Describes how to customize Tivoli Workload Scheduler for z/OS.
- *IBM Tivoli Workload Scheduler for z/OS Diagnosis Guide and Reference*, SC32-1261
Provides information to help diagnose and correct possible problems when using the product.
- *IBM Tivoli Workload Scheduler for z/OS General Information*, SC32-1256
Describes the benefits of the entire Tivoli Workload Scheduler suite.
- *IBM Tivoli Workload Scheduler for z/OS Getting Started*, SC32-1262
Describes concepts and terminology of IBM Tivoli Workload Scheduler for z/OS and provides instructions to get you working with the product as soon as possible.
- *IBM Tivoli Workload Scheduler for z/OS Installation Guide*, SC32-1264
Explains how to install Tivoli Workload Scheduler for z/OS.
- *IBM Tivoli Workload Scheduler for z/OS Licensed Program Specifications*, GI11-4208
Provides planning information about Tivoli Workload Scheduler for z/OS.
- *IBM Tivoli Workload Scheduler for z/OS Managing the Workload*, SC32-1263

Explains how to plan and schedule the workload and how to control and monitor the current plan.

- *IBM Tivoli Workload Scheduler for z/OS Memo to Users*, GI11-4209
Provides a summary of changes for the current release of the product.
- *IBM Tivoli Workload Scheduler for z/OS Messages and Codes*, SC32-1267
Explains messages and codes.
- *IBM Tivoli Workload Scheduler for z/OS Program Directory*, GI11-4203
Provided with the Tivoli Workload Scheduler for z/OS installation tape, describes all of the installation materials and gives installation instructions specific to the product release level or feature number.
- *IBM Tivoli Workload Scheduler for z/OS Programming Interfaces*, SC32-1266
Provides information to write application programs.
- *IBM Tivoli Workload Scheduler for z/OS Quick Reference*, SC32-1268
Provides a quick and easy consultation reference to operate the product.

Softcopy Collection Kit for IBM Tivoli Workload Scheduler for z/OS

All the books in the library, except the licensed publications, are available in displayable softcopy form in the following collection kits:

- IBM Online Library Omnibus Edition OS/390[®] Collection Kit, SK2T-6700.
- IBM Online Library z/OS Software Products Collection Kit, SK2T-4270.

You can read the softcopy books on CD-ROMs using these IBM licensed programs:

- Softcopy Reader
- BookManager[®] READ/2
- BookManager READ/DOS
- BookManager READ/6000

All the BookManager programs need a personal computer equipped with a CD-ROM disk drive (capable of reading disks formatted in the ISO 9660 standard) and a matching adapter and cable. For additional hardware and software information, refer to the documentation for the specific BookManager product you are using.

Updates to books between releases are provided in softcopy only.

Related publications

The *Tivoli Software Glossary* includes definitions for many of the technical terms related to Tivoli software. The *Tivoli Software Glossary* is available, in English only, at the following Web site:

<http://publib.boulder.ibm.com/tividd/glossary/termsmst04.htm>

Accessing Publications Online

The documentation CD contains the publications that are in the product library. The format of the publications is PDF, HTML, or both. To access the publications using a Web browser, open the `infocenter.html` file. The file is in the appropriate publications directory on the product CD.

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli Software Information Center Web site. The Tivoli Software Information Center is located at the following Web address:

<http://publib.boulder.ibm.com/tividd/td/tdprodlist.html>

Click the Tivoli Workload Scheduler link to access the product library.

Note: If you print PDF documents on other than letter-sized paper, select the **Fit to page** check box in the **Adobe Acrobat Print** window. This option is available when you click **File** → **Print**. **Fit to page** ensures that the full dimensions of a letter-sized page print on the paper that you are using.

Ordering Publications

You can order many Tivoli publications online at the following Web site:

<http://www.elink.ibm.com/public/applications/publications/cgibin/pbi.cgi>

You can also order by telephone by calling one of these numbers:

- In the United States: 800-879-2755
- In Canada: 800-426-4968

In other countries, see the following Web site for a list of telephone numbers:

<http://www.ibm.com/software/tivoli/order-lit/>

Providing Feedback about Publications

If you have comments or suggestions about Tivoli products and documentation, complete the customer feedback survey at the following Web site:

<http://www.ibm.com/software/sysmgmt/products/support>

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 additional information, see the Accessibility Appendix in the *IBM Tivoli Workload Scheduler Job Scheduling Console User's Guide*.

Contacting IBM Software Support

If you have a problem with any Tivoli product, you can contact IBM Software Support. See the *IBM Software Support Guide* at the following Web site:

<http://techsupport.services.ibm.com/guides/handbook.html>

The guide provides information about how to contact IBM Software Support, depending on the severity of your problem, and the following information:

- Registration and eligibility
- Telephone numbers and e-mail addresses, depending on the country in which you are located
- Information you must have before contacting IBM Software Support

Conventions Used in This Manual

This manual uses several conventions for special terms and actions, operating system-dependent commands and paths.

Typeface conventions

This manual uses the following typeface conventions:

Bold

- Lowercase commands and mixed case commands (except those in all capital letters) that are otherwise difficult to distinguish from surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes, multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Tip:**, and **Operating system considerations:**)
- Column headings in a table
- Keywords and parameters in text

Commands (except those in all capital letters) are displayed in bold font.

Italic

- Citations (titles of books, diskettes, and CDs)
- Words defined in text
- Emphasis of words (words as words)
- Letters as letters
- New terms in text (except in a definition list)
- Variables and values you must provide

Monospace

- Examples, code examples, and output
- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

Code examples, output, text you must type, and system messages are displayed in monospace font.

revision bars Technical changes to the text are indicated by a vertical line to the left of the change.

Operating system-dependent variables and paths

This manual uses the UNIX[®] convention for specifying environment variables and for directory notation.

When using the Windows[®] command line, replace *\$variable* with *% variable%* for environment variables and replace each forward slash (/) with a backslash (\) in directory paths.

Note: If you are using the bash shell on a Windows system, you can use the UNIX conventions.

Part 1. Common Installation and Configuration Tasks

Chapter 1. Product Overview

Extended agents (XA) are used to extend the job scheduling functions of IBM Tivoli Workload Scheduler to other systems and applications. An extended agent is defined as a workstation that has a host and an access method.

The host is a Tivoli Workload Scheduler fault-tolerant agent (FTA) or standard agent (SA).

The access method is a program that is executed by the hosting workstation whenever Tivoli Workload Scheduler, either through its command line or the Tivoli Job Scheduling Console, needs to interact with the external system. Tivoli Workload Scheduler for Applications includes the following access methods:

- Oracle e-Business Suite access method (MCMAGENT)
- PeopleSoft access method (psagent)
- R/3 access method (r3batch)
- z/OS access method (mvsca7, mvsjes, and mvsopc)

To launch and monitor a job on an extended agent, the host executes 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.

An extended agent workstation is only a logical entity related to an access method hosted by the physical Tivoli Workload Scheduler workstation. More than one extended agent workstations can be hosted by the same Tivoli Workload Scheduler workstation and rely on the same access method. The x-agent is defined in a standard Tivoli Workload Scheduler workstation definition, which gives the x-agent a name and identifies the access method.

The next figure shows how these elements fit together in the case of a typical extended agent configuration.

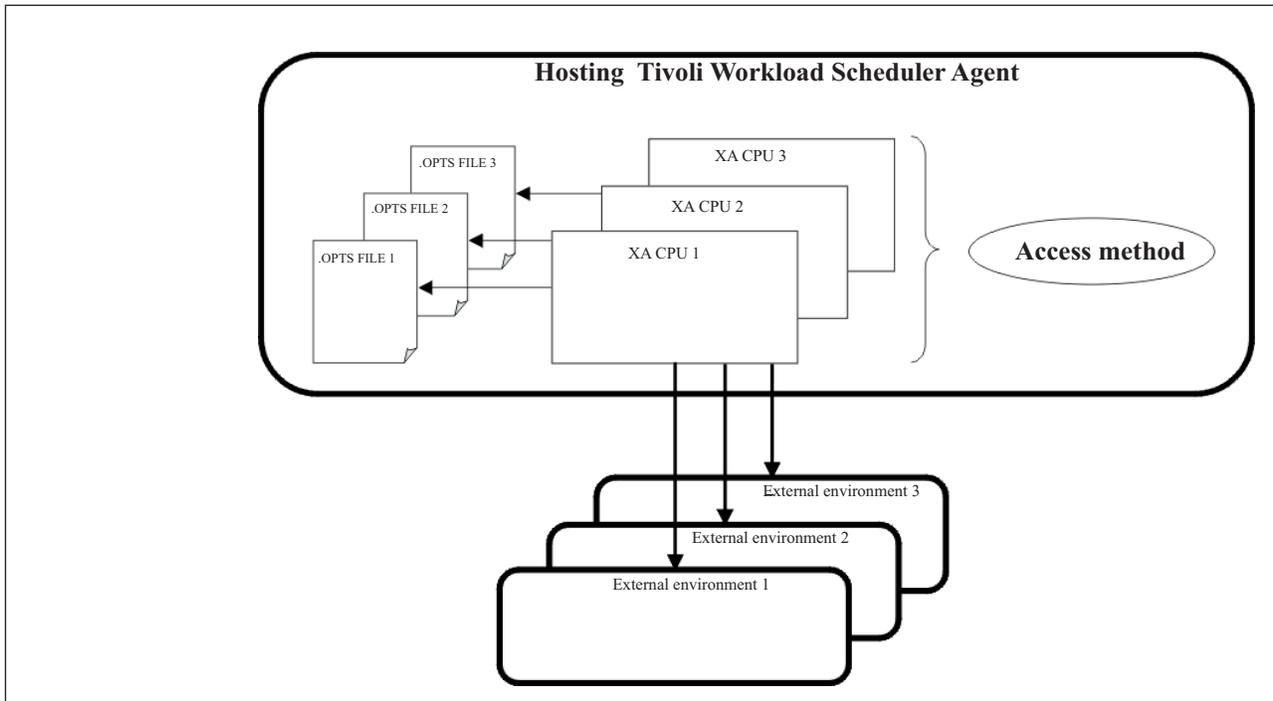


Figure 1. How Extended Agents Work.

To launch a job in an external environment, Tivoli Workload Scheduler executes 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 `<WORKSTATION_NAME>_<method_name>.opts` to determine which external environment instance it will connect 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.

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

Chapter 2. Installing IBM Tivoli Workload Scheduler for Applications

On most of the supported platforms, all the access methods of IBM Tivoli Workload Scheduler for Applications described in this guide are installed through a common installation process. Following this process, you install an instance of Tivoli Workload Scheduler for Applications -where an instance corresponds to the addition of at least one access method to an existing Tivoli Workload Scheduler installation - and select one or more access methods from the installation features.

The common installation process is available for the following platforms when supported by an access method:

- Microsoft® Windows NT or 2000
- IBM AIX®
- HP-UX
- Sun Operating Environment
- Linux for Intel
- Linux/390

For the following platforms, the installation process remains as described in the documentation of the individual methods in the next chapters:

- Compaq Tru64
- SGI Irix
- IBM Sequent® Dynix

The following table lists what platforms are supported by this version of IBM Tivoli Workload Scheduler for Applications:

Table 1. Supported Platforms

Platform	Access Method				
	z/OS (Gateway)	z/OS (Transaction Server)	R/3	PeopleSoft	Oracle eBusiness Suite
IBM OS/390 V2R10 and higher	X				
IBM z/OS V1R1 and higher	X				
IBM AIX Versions 4.3.3, 5.1, 5.2		X	X		X
IBM Sequent Dynix Version 4.5.1		X			
Solaris Operating Environment Versions 7, 8, 9		X	X		X
Hewlett Packard PA-RISC Versions 11.0, 11i		X	X		X

Table 1. Supported Platforms (continued)

Platform	Access Method				
	z/OS (Gateway)	z/OS (Transaction Server)	R/3	PeopleSoft	Oracle eBusiness Suite
SuSE Linux Enterprise Server Version 7 for OS/390 and zSeries™ (kernel 2.4)		X	X		
RedHat Linux for OS/390 and zSeries Version 7		X	X		
RedHat Linux (Intel) Versions 7.2, 7.3			X		
SuSE Linux (Intel) Enterprise Server Version 8			X		
Microsoft Windows NT® Version 4.0 with Service Pack 6a		X	X	X	
Microsoft Windows 2000 with Service Pack 3 Professional, Server, Advanced		X	X	X	
Compaq Tru64 Version 5.1		X	X		
SGI Irix Version 6.5.x		X			

How to Install

You can choose from three alternative ways to install Tivoli Workload Scheduler for Applications:

- Using the InstallShield Multi-Platform (ISMP) wizard
- Running the InstallShield silent installation option
- Using the Software Distribution component of Tivoli Configuration Manager

Before you install the product by any of these methods, make sure you meet the following conditions:

- You must have a valid installation of IBM Tivoli Workload Scheduler Version 7.0 or higher on your computer before you install any of the access methods.
- You must have root (Unix) or Administrator (Windows) rights to run the installation.
- You must stop all Tivoli Workload Scheduler services before you start the installation process.
- For the access method for PeopleSoft: Installing this method pre-requires that the corresponding PeopleSoft environment(s) be already installed and fully accessible from the workstation where you are installing.
- For the access method for Oracle eBusiness Suite: The Oracle e-Business Suite server is required on the same physical node as the access method and the Tivoli Workload Scheduler host.

- Ensure that the installation process is not already running on the workstation. This scenario could happen in a network where more telnet connections are opened on the same Unix computer by authorized users to remotely drive the installation of an access method. Run the following command to find whether the installation process is already active on the computer where you want to install:

```
ps -ef | grep setup.bin
```

By the same token, if your workstation runs more instances of Tivoli Workload Scheduler, restrain from running the installation process on more than one instance at a time.

- Before installing additional instances of the product, or updating existing ones, make sure that the Option Editor is not running or the installation may fail.

See Chapter 31, “Installed Files”, on page 183 for a list of the files you should find on your computer after installation.

Using the Installation Wizard

To be able to use the ISMP installation wizard on your computer, your operating system must be able to run Java™ Runtime Environment (JRE) Version 1.3.1. Some platforms require maintenance levels to support JRE 1.3.1. Refer to the following table to learn if your operating system requires to be patched or not. For the supported platforms not included in this table, no additional patches are required.

Table 2. Patches Required to Run JRE 1.3.1

Operating System	Patch Required to Run JRE 1.3.1
IBM AIX Version 4.3.3	AIX 4330–10 Recommended Maintenance Level.
IBM AIX Version 5.1	<p>AIX 5100–02 Recommended Maintenance Level. This requires that APAR IY19375 be previously applied and committed.</p> <p>If you are using one of the supported non-UTF8 CKJ locales, one of the following filesets (available on the AIX 5.1 base CD) is required:</p> <ul style="list-style-type: none"> • X11.fnt.ucs.ttf (for ja_JP or Ja_JP) • X11.fnt.ucs.ttf_CN (for zh_CN or Zh_CN) • X11.fnt.ucs.ttf_KR (for ko_KR) • X11.fnt.ucs.ttf_TW (for zh_TW or Zh_TW) <p>For Japanese users: if you are using Japanese Input Method, you may apply the following PTFs to avoid some Input Method-related problems.</p> <ul style="list-style-type: none"> • jkit.Wnn6.base 2.2.0.2 (PTF U479697 or APAR IY22917) (Wnn6 user only) • X11.motif.lib 5.1.0.15 (PTF U479604 or APAR IY22933) (AIX 5.1 user only)
Solaris Operating Environment Versions 7, 8, 9	Patch Clusters dated January 2002 or later.

Table 2. Patches Required to Run JRE 1.3.1 (continued)

Operating System	Patch Required to Run JRE 1.3.1
Hewlett Packard PA-RISC Versions 11.0, 11i	<p>Operating system patches should be installed before you install the product. To determine which patches have been installed on your computer, login as root and run:</p> <pre>/usr/sbin/swlist -l product</pre> <p>To support JRE 1.3.1.05, you should have:</p> <ul style="list-style-type: none"> • For HP-UX 11.0, Quality Pack QPK1100 • For HP-UX 11i, Quality Pack GOLDQPK11i

To install one or more of the access methods of Tivoli Workload Scheduler for Applications on your computer, do the following:

1. Stop all Tivoli Workload Scheduler services.
2. From the product CD or from an image run the installation command. This can be:
 - setup.bin on Unix
 - setup.exe on Windows

The Welcome window is displayed.

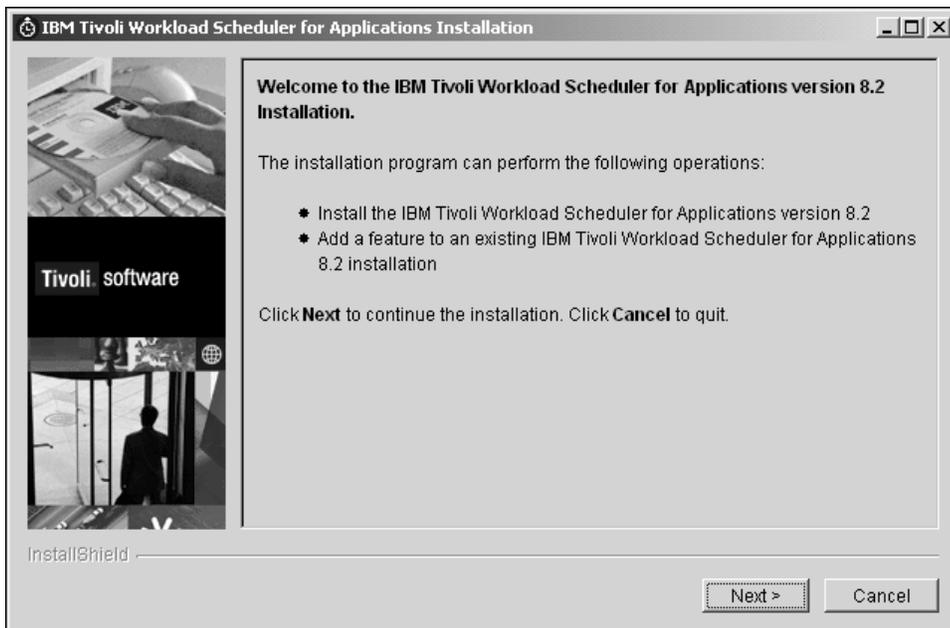


Figure 2. Welcome Window.

3. Click **Next** to continue. The Software License Agreement window is displayed.



Figure 3. Software License Agreement Window.

4. Select the radio button to accept the agreement and click **Next** to continue. The Choose a Setup Type window is displayed.

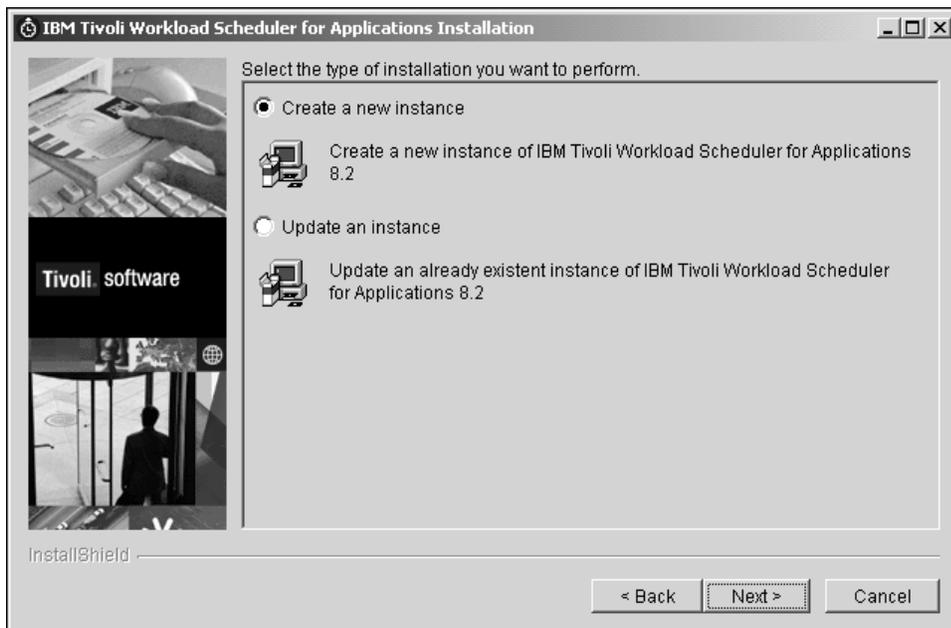


Figure 4. Choose a Setup Type Window

5. Choose whether you are performing a new installation of Tivoli Workload Scheduler for Applications or if you are updating an existing one. If you are installing an additional access method, select **Update an instance**.
 - If you choose **Create a new instance**, the installation program detects and lists all installations of Tivoli Workload Scheduler on your computer.

- If you choose **Update an instance**, the installation program detects and lists all installations of Tivoli Workload Scheduler for Applications on your computer.

Click **Next** to continue.

If you selected **Create a new instance**, a window similar to Figure 5 is displayed.

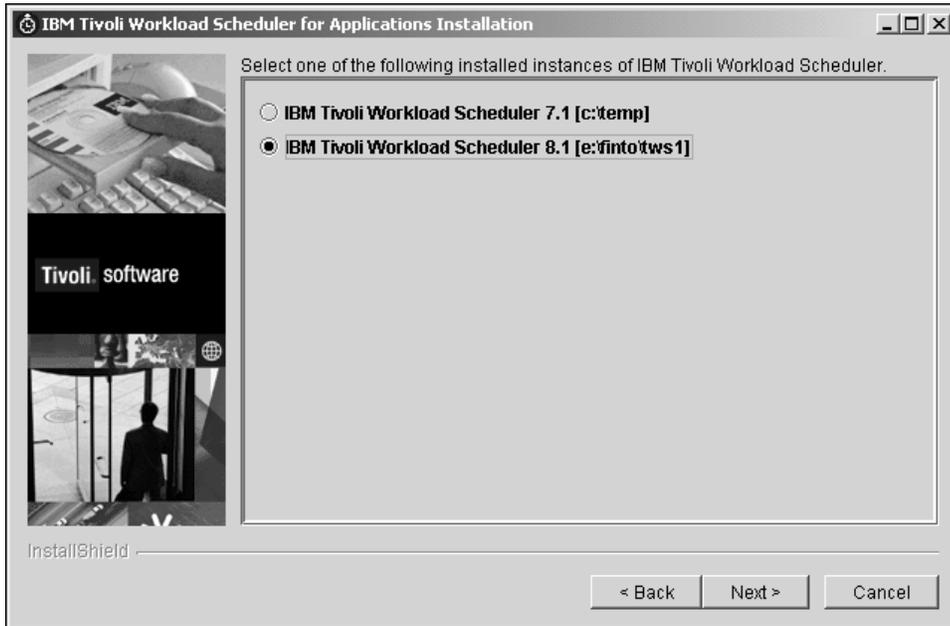


Figure 5. Installed Instances of Tivoli Workload Scheduler Window.

The installation program searches the system and lists all the existing instances of Tivoli Workload Scheduler.

Notes:

- a. If the discovery process does not find existing instances, a window is displayed where you can enter the installation path and the name of an instance.
- b. If you are on Unix and have multiple installations of Tivoli Workload Scheduler versions 7.0 or 8.1 that were not properly installed under a `/home/TWSuser` home directory, the window randomly lists only one of such instances and names it `<unknown>`. Of these instances, this will be the only one you can mark for installation. See Table 24 on page 181 for details.

If you selected **Update an instance**, a window similar to Figure 6 on page 11 is displayed.

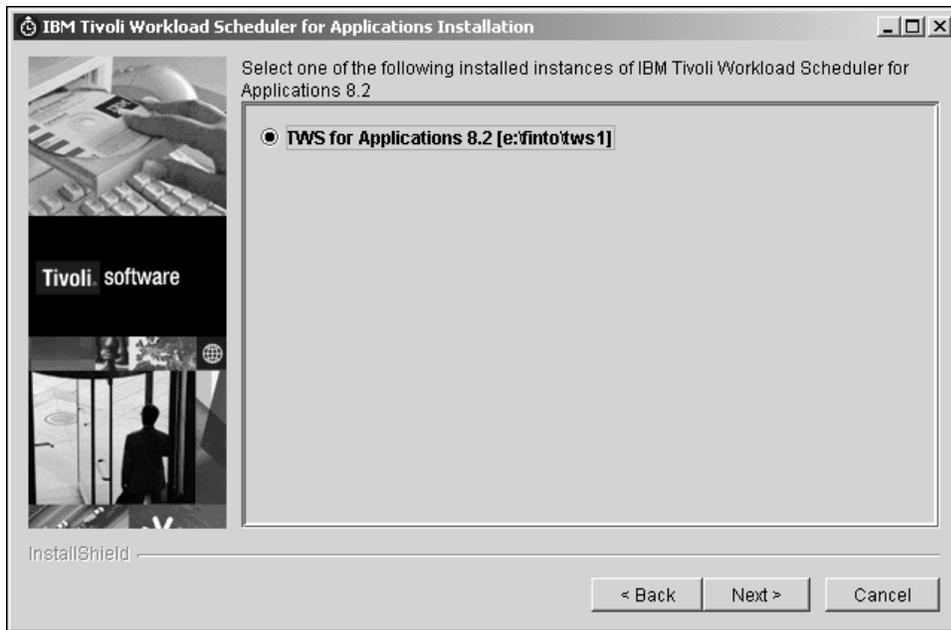


Figure 6. Installed Instances of Tivoli Workload Scheduler for Applications Window.

The installation program searches the system and lists all the existing instances of Tivoli Workload Scheduler for Applications.

6. Select one or more instances and click **Next** to continue. A window that lists all the access methods featured in Tivoli Workload Scheduler for Applications is displayed. The list varies upon the platform on which you are installing: installations on Unix do not list the access method for PeopleSoft, whereas those on Windows do not list the access method for Oracle.

This example is for an installation on Windows.



Figure 7. The Select the Tivoli Workload Scheduler for Applications Features Window on MS Windows.

This example is for an installation on Unix.

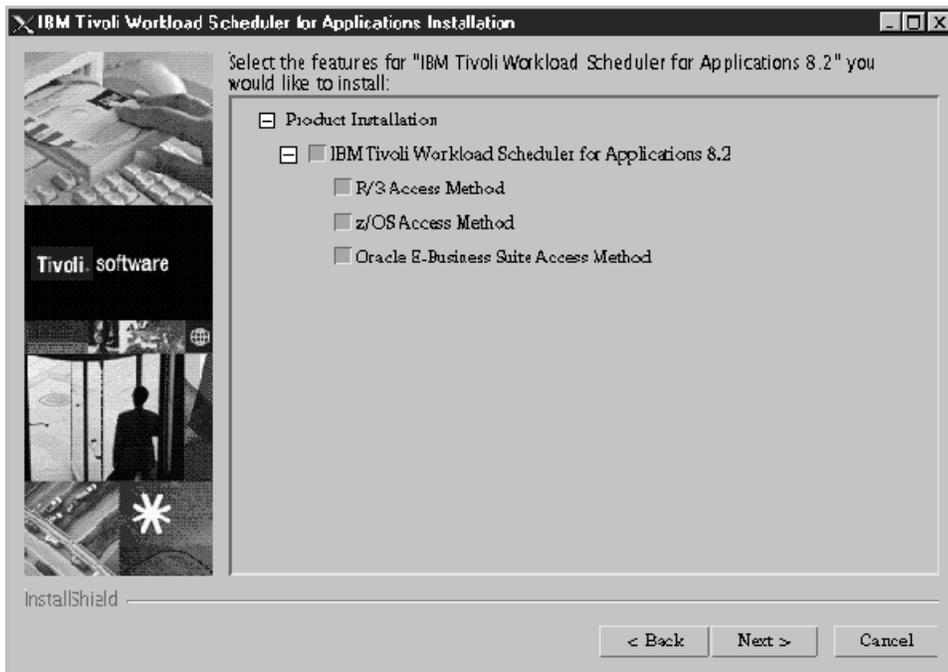


Figure 8. The Select the Tivoli Workload Scheduler for Applications Features Window on Unix.

7. Select the access method or methods that you want to install on your computer and click **Next** to continue. Depending on how many and which access methods you selected, one or more additional windows request for access method-specific information. This can be:
 - For all access methods: the name of the Tivoli Workload Scheduler user who will be running the selected method. This can be the same name that was defined when Tivoli Workload Scheduler was installed.
 - For the PeopleSoft access method: the version of the PeopleTools environment and one of the following:
 - For PeopleSoft Versions 7.0x or 7.5x, the path to the COBOL runtime directory (usually `<PS_HOME>\cb1bin`).
 - For PeopleSoft Version 8.1x, the path to the PeopleSoft External APIs (usually `<EXT_API_DIR>\bin\client\winx86_extapi`).
 - For PeopleSoft Version 8.4x, the path to the PeopleSoft client directory (usually `<PS_HOME>\bin\client\winx86`).

This is the window for the R/3 Access Method.

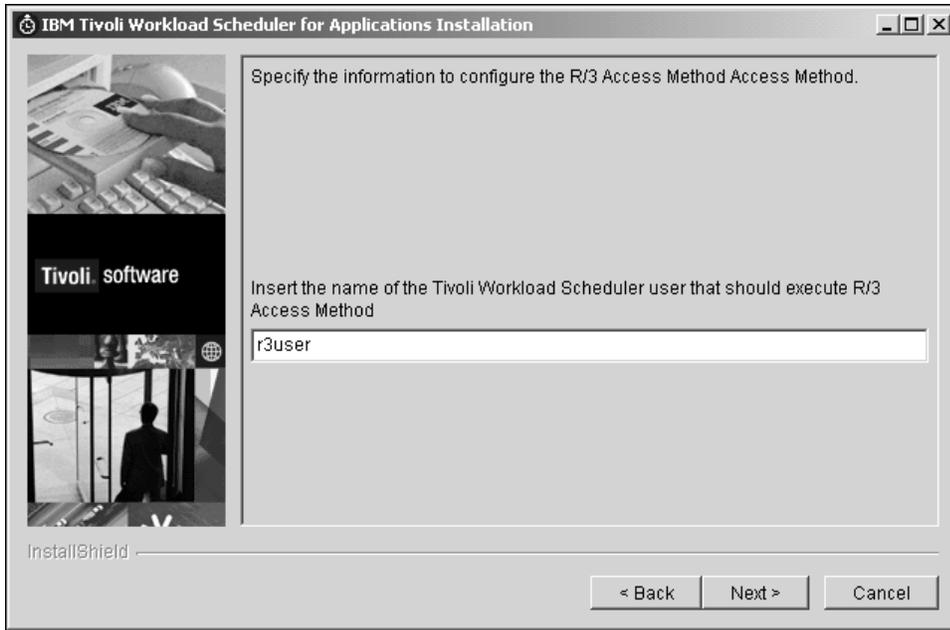


Figure 9. Additional User Information Window for the R/3 Access Method.

This is the first window for the PeopleSoft Access Method.

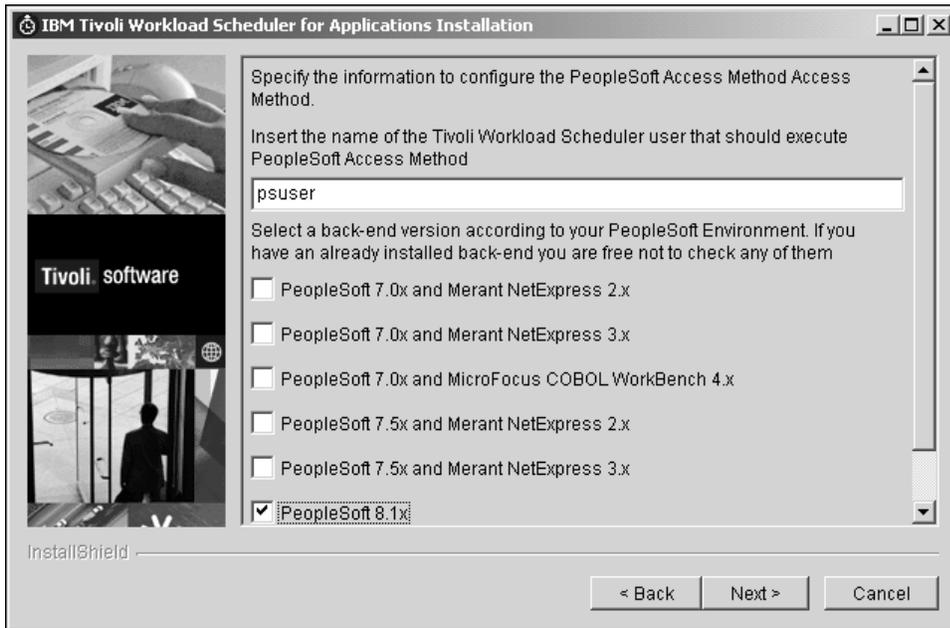


Figure 10. First Additional User Information Window for the PeopleSoft Access Method.

This is the second window for the PeopleSoft Access Method.

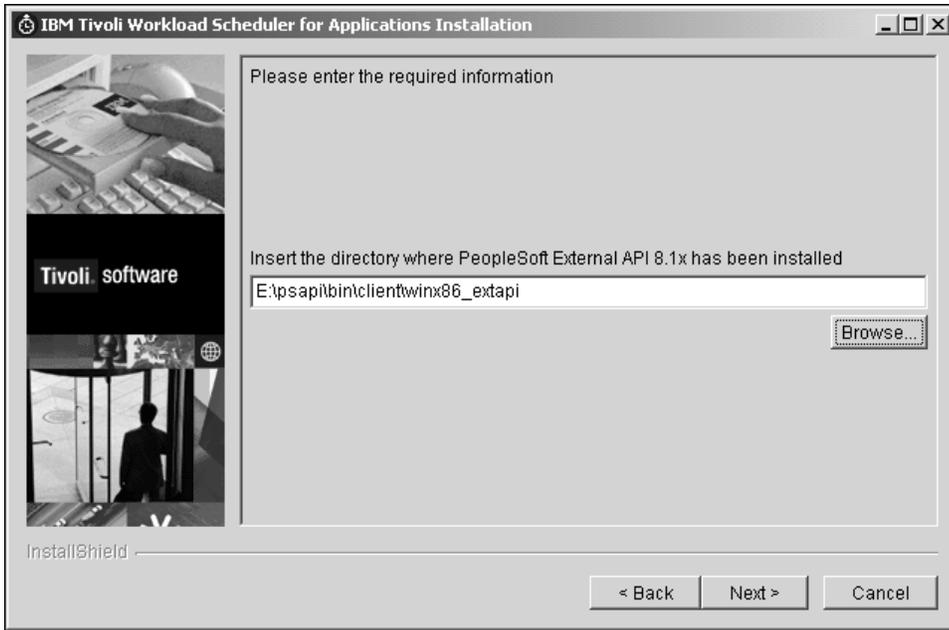


Figure 11. Second Additional User Information Window for the PeopleSoft Access Method.

This is the window for the z/OS Access Method.



Figure 12. Additional User Information Window for the z/OS Access Method.

8. Enter the required information and click **Next** to continue. A summary information window is displayed. Figure 13 on page 15 shows an example of the summary information for the installation of the access method for R/3.



Figure 13. Installation Summary Window.

9. Click **Next** to continue. The installation process begins to install the selected access methods on your computer. A window showing the installation progress is displayed.

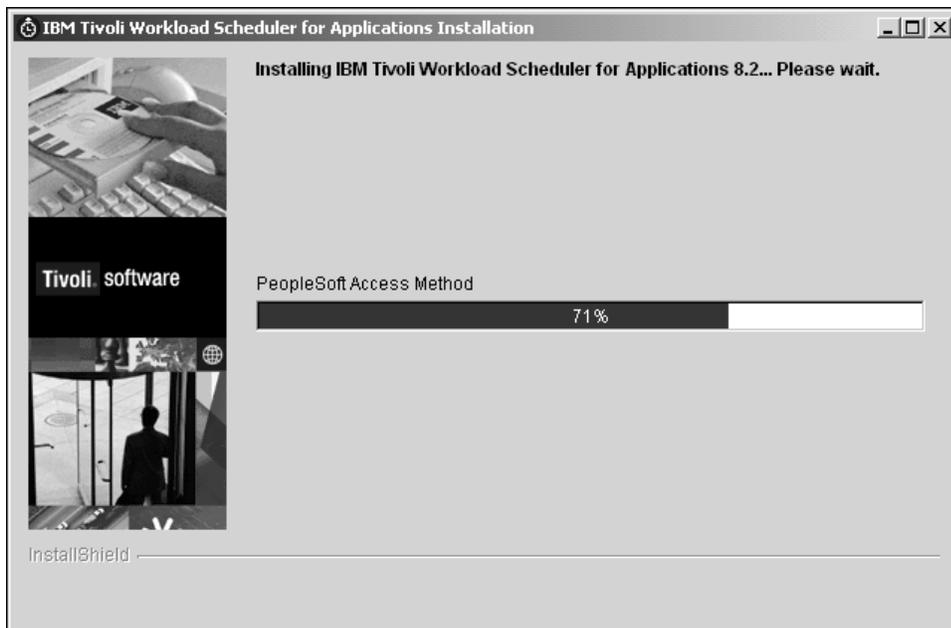


Figure 14. The Installation Progress Window.

When the installation has completed, the following window is displayed asking if you want to run the Option Editor. This is a Java-based files editor which enables you to define options for the access methods you installed. If you opt for not running the Option Editor now, you can do it anytime later.

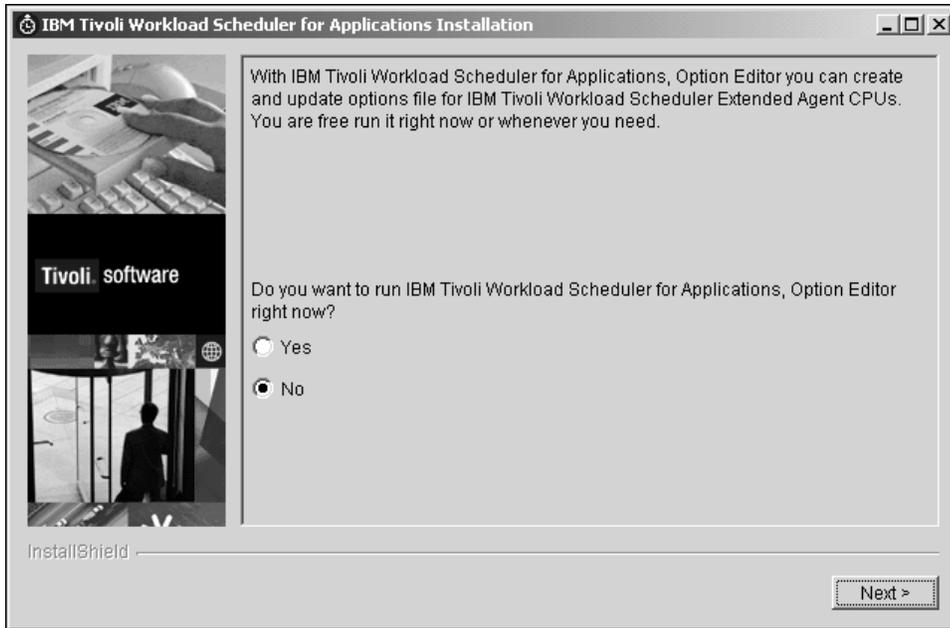


Figure 15. Option Editor Launch Window

10. Click on the appropriate radio button if you want to immediately run the Option Editor or not and select **Next** to continue.
If you click **Yes**, the Option Editor is displayed and you can set your options for the methods you installed. See Chapter 3, “Setting Options with the Option Editor”, on page 25 to learn how to use the Option Editor.
When you have finished with the Option Editor or if you had selected **No** in the previous window, the installation exit window is displayed.

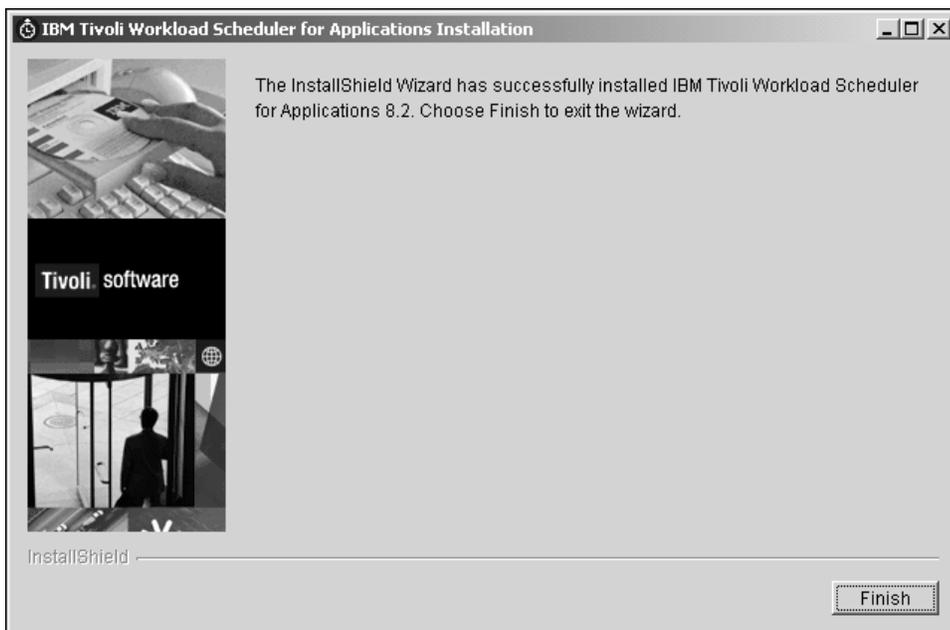


Figure 16. The Installation Wizard Exit Window.

11. Select **Finish** to exit the installation wizard.

The installation log named TWS4APPS.LOG is saved in the maestro directory.

Using Silent Installation

This method provides the option to install the product without using the ISMP installation wizard. To install by this method, you prepare an options file beforehand and specify its name with the installation command.

The use of the silent installation method enables you to install up to all the access methods simultaneously.

The installation log named TWS4APPS.LOG is saved in the maestro directory.

Format: To install Tivoli Workload Scheduler for Applications with the silent method, use the following command:

- On Windows

```
setup.exe -silent -options options_file_path_and_name
```
- On Unix

```
setup.bin -silent -options options_file_path_and_name
```

Parameters:

-silent Specifies that installation is not driven by a user interface but by an options file.

-options *options_file_path_and_name*

Provides the path and the name of the file that contains the installation options. The options file can be any text file with the name and extension you choose. Options are written individually per line.

A template for running a silent installation is provided with the product. The template is named `silent.txt` and is available in the `RESPONSE_FILE` directory in the product CD-Rom.

The following table lists the options you can use to drive the installation:

Table 3. Options Required for Silent Installation.

Option	Description	Value
-P TWS4APPS.installLocation	The path where you want to install Tivoli Workload Scheduler for Applications. This option is mandatory.	An absolute path.
-P TWS4APPS.name	A name for this instance of Tivoli Workload Scheduler for Applications. This option is mandatory.	A name. Enclose within double quotes if the name contains a blank.
-P MCMAGENT_FEATURE.active	Specifies that the access method for Oracle eBusiness Suite is to be installed.	True or False.
-W LJUSER_MCMAGENT_INPUT_BEAN.LJUSER	The name of the Tivoli Workload Scheduler user who is to execute the access method for Oracle eBusiness Suite. This option is mandatory if you are installing this method.	A Tivoli Workload Scheduler user name. Enclose within double quotes if the name contains a blank.

Table 3. Options Required for Silent Installation. (continued)

Option	Description	Value
-P R3BATCH_FEATURE.active	Specifies that the access method for R/3 is to be installed.	True or False.
-W LJUSER_R3BATCH_INPUT_BEAN.LJUSER	The name of the Tivoli Workload Scheduler user who is to execute the access method for R/3. This option is mandatory if you are installing this method.	A Tivoli Workload Scheduler user name. Enclose within double quotes if the name contains a blank.
-P ZOSAGENT_FEATURE.active	Specifies that the access method for z/OS is to be installed.	True or False.
-W LJUSER_ZOSAGENT_INPUT_BEAN.LJUSER	The name of the Tivoli Workload Scheduler user who is to execute the access method for z/OS. This option is mandatory if you are installing this method.	A Tivoli Workload Scheduler user name. Enclose within double quotes if the name contains a blank.
-P PSAGENT_FEATURE.active	Specifies that the access method for PeopleSoft is to be installed.	True or False.
-W LJUSER_PSAGENT_INPUT_BEAN.LJUSER	The name of the Tivoli Workload Scheduler user who is to execute the access method for PeopleSoft. This option is mandatory if you are installing this method.	A Tivoli Workload Scheduler user name. Enclose within double quotes if the name contains a blank.

Table 3. Options Required for Silent Installation. (continued)

Option	Description	Value
-W LJUSER_PSAGENT_INPUT_BEAN.BEND_TYPE	Specifies the version of the PeopleTools environment. This option is mandatory if you are installing this method.	It can be one or more of the following: PS70MR2 PeopleTools 7.0x and Merant NetExpress 2.x PS70MR3 PeopleTools 7.0x and Merant NetExpress 3.x PS70WB4 PeopleTools 7.0x and MF COBOL Workbench 4.x PS75MR2 PeopleTools 7.5x and Merant NetExpress 2.x PS75MR3 PeopleTools 7.5x and Merant NetExpress 3.x PS81 PeopleTools 8.1x PS84 PeopleTools 8.4x Multiple entries must be separated by semi-colons.
-W PS70MR2_CBLDIR.DIRNAME	The path of the COBOL directory for PeopleTools 7.0x and Merant NetExpress 2.x. Use if you specified this environment.	The complete path to this directory, usually <PS_HOME>\cblbin.
-W PS70MR3_CBLDIR.DIRNAME	The path of the COBOL directory for PeopleTools 7.0x and Merant NetExpress 3.x. Use if you specified this environment.	The complete path to this directory, usually <PS_HOME>\cblbin.
-W PS70WB4_CBLDIR.DIRNAME	The path of the COBOL directory for PeopleTools 7.0x and MF COBOL Workbench 4.x. Use if you specified this environment.	The complete path to this directory, usually <PS_HOME>\cblbin.
-W PS75MR2_CBLDIR.DIRNAME	The path of the COBOL directory for PeopleTools 7.5x and Merant NetExpress 2.x. Use if you specified this environment.	The complete path to this directory, usually <PS_HOME>\cblbin.
-W PS75MR3_CBLDIR.DIRNAME	The path of the COBOL directory for PeopleTools 7.5x and Merant NetExpress 3.x. Use if you specified this environment.	The complete path to this directory, usually <PS_HOME>\cblbin.

Table 3. Options Required for Silent Installation. (continued)

Option	Description	Value
-W PS81_APIDIR.DIRNAME	The path of the External APIs directory for PeopleTools 8.1x. Use if you specified this environment.	The complete path to this directory, usually <code><EXT_API_DIR>\bin\client\winx86_extapi.</code>
-W PS84_APIDIR.DIRNAME	The path of the PeopleSoft client directory for PeopleTools 8.4x. Use if you specified this environment.	The complete path to this directory, usually <code><PS_HOME>\bin\client\winx86.</code>

Examples: The following example uses an options file named `silent_options.txt` to silently install Tivoli Workload Scheduler for Applications on a Windows computer:

```
setup.exe -silent -options c:\tmp\silent_options.txt
```

The following example uses an options file named `silent_options.txt` to silently install Tivoli Workload Scheduler for Applications on a Unix computer with no X-Server available:

```
setup.bin -silent -options /tmp/silent_options.txt
```

The following example is a file of options used to drive the installation of Tivoli Workload Scheduler for Applications on a Windows computer. The access methods for R/3 and for PeopleSoft are defined as installation features.

```
# INSTALL LOCATION and INSTANCE NAME
-P TWS4APPS.installLocation=C:\win32app\maestro
-P TWS4APPS.name=SILENT
# R3BATCH SETTINGS
-P R3BATCH_FEATURE.active=true
-W LJUSER_R3BATCH_INPUT_BEAN.LJUSER=twuser
# PSAGENT SETTINGS
-P PSAGENT_FEATURE.active=true
-W LJUSER_PSAGENT_INPUT_BEAN.LJUSER=tw
-W LJUSER_PSAGENT_INPUT_BEAN.BEND_TYPE=PS70MR2;PS75MR3
-W PS70MR2_CBLDIR.DIRNAME=M:\PSDIRS\PS70Merant2
-W PS75MR3_CBLDIR.DIRNAME=N:\PSDIRS\PS75Merant3
```

Using Tivoli Configuration Manager

Every access method can be installed by distributing a software package block (SPB) using the Tivoli Desktop or the `winstsp` command of the Software Distribution component of IBM Tivoli Configuration Manager, Version 4.2. The distribution and installation of the Software Package Blocks can be carried through computers running the Tivoli Management Framework.

Deploying Tivoli Workload Scheduler for Applications through this packaging format provides the following set of benefits:

- Deploy not only files and directories, but also environment elements such as INI, registry, icons, and more.
- UNDO versus REMOVE of each of the above built-in elements in the software package.
- Use of variables (built-in, target-specific or predefined) in the package.
- Use of conditioning in the software package through specific sections for specific platforms.

- Unlimited use of generic user-programs, for different execution phases, and custom management of program return codes.
- Reboot or checkpoint restart during the execution of the SP actions

Whith this method you can install only one access method per software package block. If you use multiple SPBs to install more access methods, be careful to use different names for the packages.

If you choose to drive the installation process from the Tivoli Desktop, and your Tivoli Workload Scheduler engine in the target workstation is Version 8.2, make sure you know the name of the installation directory of the engine before you start.

To be installed by software distribution, all SPBs of Tivoli Workload Scheduler for Applications require the following options :

Table 4. Options Required to Install all SPBs of Tivoli Workload Scheduler for Applications by Software Distribution.

Option	Description
INSTALL_DIR	The path to the maestro directory (which contains methods)
LJUSER	The name of the Tivoli Workload Scheduler user who is to execute the access method.

Additionally, the SPB for the PeopleSoft access method requires the following options:

Table 5. Additional Options Required to Install the access method for PeopleSoft SPB by Software Distribution.

Option	Description
INSTALL_BEND	Specifies the version of the PeopleTools environment. It can be one or more of the following: PS70MR2 PeopleTools 7.0x and Merant NetExpress 2.x PS70MR3 PeopleTools 7.0x and Merant NetExpress 3.x PS70WB4 PeopleTools 7.0x and MF COBOL Workbench 4.x PS75MR2 PeopleTools 7.5x and Merant NetExpress 2.x PS75MR3 PeopleTools 7.5x and Merant NetExpress 3.x PS81 PeopleTools 8.1x PS84 PeopleTools 8.4x Multiple entries must be separated by semi-colons.
PS70MR2_CBLDIR	The path of the COBOL directory for PeopleTools 7.0x and Merant NetExpress 2.x. Use if you specified this environment.
PS70MR3_CBLDIR	The path of the COBOL directory for PeopleTools 7.0x and Merant NetExpress 3.x. Use if you specified this environment.
PS70WB4_CBLDIR	The path of the COBOL directory for PeopleTools 7.0x and MF COBOL Workbench 4.x. Use if you specified this environment.

Table 5. Additional Options Required to Install the access method for PeopleSoft SPB by Software Distribution. (continued)

Option	Description
PS75MR2_CBLLDIR	The path of the COBOL directory for PeopleTools 7.5x and Merant NetExpress 2.x. Use if you specified this environment.
PS75MR3_CBLLDIR	The path of the COBOL directory for PeopleTools 7.5x and Merant NetExpress 3.x. Use if you specified this environment.
PS81_APIDIR	The path of the External APIs directory for PeopleTools 8.1x. Use if you specified this environment.
PS84_APIDIR	The path of the PeopleSoft client directory for PeopleTools 8.4x. Use if you specified this environment.

The product includes also an SPB containing a Java Virtual Machine. This is useful to install the Option Editor with this method. Be aware, however, that this Java Virtual Machine cannot replace the Java Virtual Machine required for running the PeopleSoft product.

To install a software package, you must perform the following actions, using either the command line or the Tivoli Desktop:

1. Import the SPB in a Software Package Profile. Do this for the SPBs of:
 - The access methods that you want to install
 - The Java Virtual Machine (JVM)
 - The Option Editor (to execute, requires that the other two SPBs be already installed)
2. Install the Software Package Profile on the endpoints managed by the Tivoli Framework.

The following example shows how to install the SPB of the R/3 access method on a Windows workstation.

1. Import the SBP file TWS4APPS_R3BATCH_I386NT.SPB in a Tivoli Framework Software Package Profile named, for example, TWS4APPS_R3BATCH_I386NT.8.2. Use the following command:

```
wimspso -c PROFILE_MANAGER -f <SOURCE_PATH>/TWS4APPS_R3BATCH_I386NT.SPB -t build
-p <DEST_PATH_LOCAL>/TWS4APPS_R3BATCH_I386NT.SPB TWS4APPS_R3BATCH_I386NT.8.2
```

With this command, the SOURCE_PATH/TWS4APPS_R3BATCH_I386NT.SPB file is imported in a Software Package Profile named TWS4APPS_R3BATCH_I386NT.8.2 that belongs to a Tivoli Framework PROFILE_MANAGER profile manager.

TWS4APPS_R3BATCH_I386NT.8.2 must refer to <DEST_PATH_LOCAL>/TWS4APPS_R3BATCH_I386NT.SPB as the local file holding the needed information.

Note: In order to install an additional instance of this SPB on the same workstation, you have to rename the SPB.

2. Install the TWS4APPS_R3BATCH_I386NT.8.2 Software Package Profile on a computer with hostname epw2000 (which must be also defined as a Tivoli Framework endpoint). Use the following command:

```
winstsp -D INSTALL_DIR=TWShome -D LJUser=TWUser TWS4APPS_R3BATCH_I386NT.8.2
epw2000
```

When the installation log, in this case a file named `$BINDIR/./swdis/work/TWS4APPS_R3BATCH_I386NT.log`, shows an Operation Successful message, the installation process may be considered as completed.

The following example shows how to install the SPB of the JVM for AIX on an AIX system. Note that a different JVM software package exists for each platform.

1. Import the software package named `JRE131_AIX.SPB` into a Software Package Profile named `JRE131AIX.8.2`. Use the following command:

```
wimpspo -c PROFILE_MANAGER -f <SOURCE_PATH>/JRE131_AIX.SPB -t build
-p <DEST_PATH_LOCAL>/JRE131_AIX.SPB
JRE131AIX.8.2
```

2. Install the `JRE131AIX.8.2` Software Package Profile on a computer with hostname `epAIX`. Use the following command:

```
winstsp -D INSTALL_DIR=TWShome -D LJUser=TWSuser JRE131AIX.8.2 epAIX
```

The following example shows how to install the SPB of the Option Editor on the same AIX system. Note that the SPB of the Option Editor is the same for all platforms.

1. Import the software package named `OPTED.SPB` into a Software Package Profile named `OPTED.8.2`. Use the following command:

```
wimpspo -c PROFILE_MANAGER -f <SOURCE_PATH>/OPTED.SPB -t build
-p <DEST_PATH_LOCAL>/OPTED.SPB
OPTED.8.2
```

2. Install the `OPTED.8.2` Software Package Profile on a computer with hostname `epAIX`. Use the following command:

```
winstsp -D INSTALL_DIR=TWShome -D LJUser=TWSuser OPTED.8.2 epAIX
```

Uninstalling the Product

Uninstalling Tivoli Workload Scheduler for Applications from a workstation requires that you follow the same procedure you used at installation time:

- If you installed through ISMP, uninstall by an ISMP-compatible method.
- If you installed through software distribution, uninstall by the same method.

If you plan to uninstall the access method for Oracle eBusiness suite, be aware that you will have to complete manually the removal of the product. See “Completing the Removal of the Access Method for Oracle e-Business Suite” on page 24.

Under certain conditions, when you fully uninstall the product, a directory named `_xauninst` may be left over by the uninstallation process. In this event, remove it by hand.

Removing an Installation via ISMP

To uninstall the product following an ISMP installation:

- On Windows:
 1. Select **Add/Remove Programs** in the Control Panel.
 2. Find **IBM Tivoli Workload Scheduler for Applications** (*username*).
 3. Click **Remove**. An Uninstaller window is opened where you can choose to delete the entire product or one or more of the access methods that were installed as product features.
- On Unix:
 1. Go to the `<TWShome>/methods` directory.

2. Run the following command:

```
../_xauninst/uninstaller
```

An Uninstaller window is opened where you can choose to delete the entire product or one or more of the access methods that were installed as product features.

Removing an Installation via Tivoli Configuration Manager

To uninstall the product using software distribution, either use the Tivoli Desktop or the `wremovsp` command.

To see a list of the packages installed on the computer by software distribution, use one of the two following procedures:

- Use Tivoli Inventory, included with Tivoli Configuration Manager Version 4.2, to run the query named `CM_STATUS_QUERY` and list what access method is installed on which workstation. See the Tivoli Inventory documentation for reference.
- View the Software Distribution log files on the computer. Every software distribution package has its own log file — a text file that records the installation and uninstallation history of the package. For instance, the log file on an AIX system of the R/3 access method package, `r3batch_tws_aix.log`, is stored in the `$BINDIR/./swdis/work` directory (where `$BINDIR` is an environment path of the Tivoli Framework that is initialized by running the `/etc/Tivoli/setup_env.sh` command).

See the Tivoli Software Distribution documentation for details.

Completing the Removal of the Access Method for Oracle e-Business Suite

After the uninstallation process is over, 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'
```

should become

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

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

Chapter 3. Setting Options with the Option Editor

An options file is a text file containing a set of configuration options required to run an extended agent. The options are written in the order of one per line and have the following format:

```
OPTION=VALUE
```

with no spaces around or between the keywords.

The options are used to alter the run-time properties of the access method, such as the password needed to connect to an external system (like SAP R/3) or the login name for that connection.

All access methods use two types of options files. They are:

Global Options File

A common configuration file for a particular instance of an access method whose settings affect all the extended agent workstations defined in that instance. It lists options such as LJuser (which provides the name of the user who executes the access method) and IFuser (which provides the name of the user who executes the method to retrieve job information) that are in part entered at installation time. The name of this file is `<access_method_name>.opts`.

For example, for an installation of the access method for R/3, the name of the global options file is `r3batch.opts`.

Local Options File

A configuration file that is specific to each extended agent workstation within a particular instance of an access method. Every extended agent workstation, except for z/OS, must have a local options file with its own configuration options. The name of this file is *Workstation name_<access_method_name>.opts*.

For example, if an installation of the access method for R/3 includes two extended agent workstations, CPU1 and CPU2, the names of the local options files will be respectively `CPU1_r3batch.opts` and `CPU2_r3batch.opts`.

The files must reside in the methods subdirectory of the Tivoli Workload Scheduler installation and are read when the extended agent is initialized. Options are specific to each access method. To know the options of each access method, see:

- Chapter 8, “MCMAGENT Options File Reference”, on page 51 for Oracle
- Chapter 11, “Post-installation Tasks”, on page 63 for PeopleSoft
- Chapter 20, “R/3 Access Method Options Files”, on page 117 for R/3
- Chapter 25, “Set Up and Operation”, on page 147 for z/OS

You can either write options files with a text editor of your choice or create them with the Option Editor provided with the product.

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

To start Option Editor at any other time, go to the methods subdirectory of your Tivoli Workload Scheduler installation and run the **opted.exe** (Windows) or **opted.bin** (Unix) command.

If you installed the product via Tivoli Configuration Manager, start Option Editor with one of these commands:

- *TWShome\methods\opted\opted.bat* on Windows
- *TWShome/methods/opted/opted.sh* on Unix

The OptionEditor automatically loads all the existing global and local options files. As well as editing existing files, you can create local options files for newly defined extended agent workstations.

The OptionEditor provides three possible views of an options file:

Simple

Displays the options as a sequence of tabs and lets you 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. It is the easiest way to edit options, because you only see the actual value that will be used in the file. Inherited or default values are not displayed. Mandatory options are in yellow and are marked by an asterisk (*).

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

Value The value specified in the file.

Default Value

The value used by the method if no any other value is available.

Inherited Value

The value obtained from the global options file if inheritance is supported by the method.

Actual Value

The value actually used at runtime. The order of use is: value, if available; inherited value, if supported; default value.

Mandatory options are in a yellow background.

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

The Option Editor window is composed of three main areas, as shown in Figure 17 on page 27. They are:

Options Files Tree

Shows the list of all the existing options files, gathered 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.

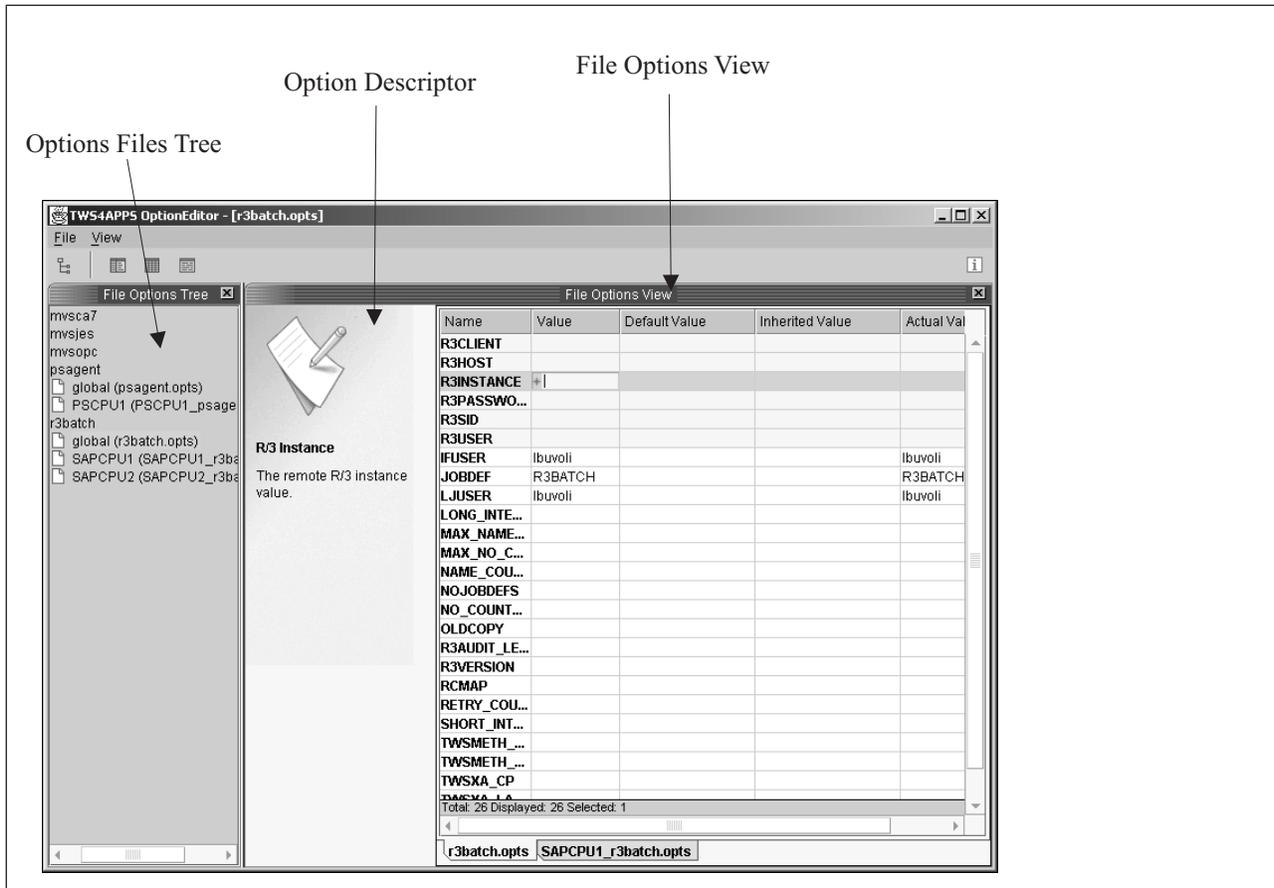


Figure 17. The Option Editor

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

You should restrict access to the Option Editor executable. If you have multiple installations of an extended agent, you can enhance security by maintaining only one copy of the Option Editor (on the master workstation for example). Using FTP, you can manage all your encrypted options files from one location.

Creating a Local Options File

While global options files are generated each time you install an instance of an access method, you can create local options files any time you desire, before or after defining an extended agent workstation. To do so with the Option Editor:

1. Click **File**→**New** in the menu bar. The New Option File window is displayed.
2. Write the name of an extended agent workstation that you have just defined or plan to define.
3. Select an access method from the installed ones in the drop-down menu. The new options file will have the name:
`<extended_agent_workstation_name>_<method_name>.opts.`
4. Start entering values for the options.

Option Value Inheritance

This property is currently available for r3batch only. It implies that local options files may inherit existing values from the same options in the global options file (r3batch.opts). In fact, for an access method the options are listed twice: once as global and once as local. If the local option does not list its own value, then the one from its global equivalent is used. On the contrary, if it exists, the local value always rules over the global one.

For example, you may want to define the same R/3 user and password for all your R/3 systems. By using this property, instead of writing the same values in every local options file, you may write them once in the global options file.

Chapter 4. Defining Extended Agent Workstations

An extended agent workstation definition is required for each entity of an access method through which Tivoli Workload Scheduler will schedule and launch jobs. Extended agent workstations are defined in the standard manner and include the name of the extended agent's host and the method name.

To launch a job on an extended agent workstation, Tivoli Workload Scheduler executes the access method, passing it information about the job. The access method communicates with the instance of the particular z/OS, R/3, PeopleSoft, or Oracle system and monitors the job through completion, writing job progress and status information to the job's standard list file.

Creating a Workstation with the Job Scheduling Console

To define an extended agent workstation with the Job Scheduling Console, follow these steps:

1. From the main window, either select the **New Workstation** icon in the top toolbar, or right click on the Tivoli Workload Scheduler engine name and select **New Workstation** from the drop down menu. The Properties - Workstation in Database window is displayed.

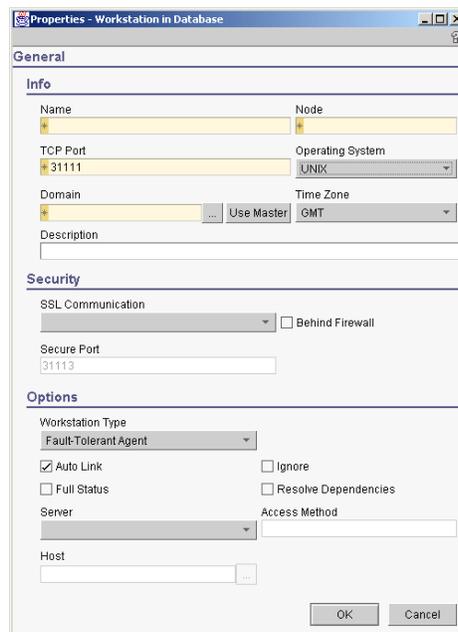


Figure 18. Defining a Workstation with the Job Scheduling Console

2. Complete the fields using the information in Table 6 on page 30.
3. Click **OK** to save and to close the window.

The following table shows how to complete the fields of the workstation properties window depending on which extended agent workstation you want to define.

Table 6. How to Complete the Fields of the Properties - Workstation in Database Window

Field	Description by Access Method			
	PeopleSoft	z/OS	Oracle eBusiness Suite	R/3
Name	<p>The name for the extended agent workstation. For all access methods (excepting the one for z/OS that is always limited to 8), the name can contain up to 8 or 16 alphanumeric characters depending if you use unexpanded or expanded databases. It can include dash and underscore characters, but it must begin with a letter.</p> <p>For all the access methods, this name must be consistent with the name of the options file associated with the workstation. That is, if the options file (that you normally create with the Option Editor) is named <mycpu>_<mymethod>.opts, then mycpu and Name must be equal.</p>			
Node	<p>null.</p>	<p>The node name or IP address of the z/OS system. Fully qualified domain names are accepted.</p>	<p>null.</p>	<p>null.</p>
TCP Port	<p>Any number other than 0.</p>	<p>The TCP address (port number) of the z/OS gateway on the z/OS system. Enter the same value as the SYSTSIN variable PORT described in Table 21 on page 141.</p>	<p>Any number other than 0.</p>	<p>Any number other than 0.</p>
Operating System	<p>Select OTHER.</p>			
Domain	<p>The name of the Tivoli Workload Scheduler domain of the host workstation. If you need help in finding it, click the ellipsis (...) for a list of existing domains. Click Use Master, if you know that the host workstation is in the same domain as the master domain manager.</p>			
Time Zone	<p>If the Tivoli Workload Scheduler network is using the Time Zone option, select the appropriate time zone from the drop down list. Leave the field blank, if this is not the case.</p>			
Description	<p>An optional free-form textual description of the workstation (up to 40 characters).</p>			
SSL Communication	<p>Not used on extended agent workstations.</p>			
Behind Firewall	<p>Not used on extended agent workstations.</p>			
Secure Port	<p>Not used on extended agent workstations.</p>			
Workstation Type	<p>Select Extended Agent.</p>			
AutoLink	<p>Not used on extended agent workstations.</p>			
Ignore	<p>Select this option only if you want Tivoli Workload Scheduler to ignore this workstation definition. This option is useful if you want to predefine job streams, jobs, and objects for a system that has not yet arrived.</p>			
Full Status	<p>Not used on extended agent workstations.</p>			
Resolve Dependencies	<p>Not used on extended agent workstations.</p>			
Server	<p>Not used on extended agent workstations.</p>			

Table 6. How to Complete the Fields of the Properties - Workstation in Database Window (continued)

Field	Description by Access Method			
	PeopleSoft	z/OS	Oracle eBusiness Suite	R/3
Access Method	psagent	<p>Depending on your job scheduling interface, one of the following:</p> <p>mvscsca7 To launch and monitor z/OS jobs via CA-7.</p> <p>mvcsjes To launch and monitor z/OS jobs via JES2 or JES3.</p> <p>mvcsopc To launch and monitor z/OS jobs via OPC or Tivoli Workload Scheduler for z/OS.</p> <p>Note: The name is case sensitive (must be in lower case) in Unix.</p>	<p>MCMAGENT</p> <p>Note: The name is case sensitive and must be in upper case.</p>	<p>r3batch</p> <p>Note: The name is case sensitive (must be in lower case) in Unix.</p>
Host	The Tivoli Workload Scheduler name of the host system. This is the Tivoli Workload Scheduler fault-tolerant agent or standard agent on which the extended agent is installed. Click the ellipsis (...) to see a list of existing hosts.			

Creating a Workstation from the Command Line

Extended agent workstation definitions can also be made in the Tivoli Workload Scheduler command line using the **composer** program. The following example shows a definition for a z/OS extended agent workstation named MVSCPU that uses the **mvcsjes** 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 mvcsjes
end

```

Consult the IBM Tivoli Workload Scheduler Reference Guide for further details about defining workstations with **composer**.

Defining Workstations for End-to-end Scheduling

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.

To this end, extended agent workstations must also be properly 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 will be actually run 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.
2. Add the same workstation definition to the database via ISPF or the Job Scheduling Console. For a description of how to define the workstation using the Job Scheduling Console, refer to the *Tivoli Job Scheduling Console User's Guide*.

Creating the CPUREC Statement

First, 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 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 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.

For further information on CPUREC, consult *Tivoli Workload Scheduler for z/OS Customization and Tuning*.

Defining the Workstation with ISPF

Figure 19 shows the ISPF definition for R3XA.

In ISPF (or in the Job Scheduling Console), define the workstation as computer automatic and then set the *FT Work station* field to **Y**. It is the CPUREC statement with the three keywords described in the previous section that provides the extended agent specification.

Note: Make sure you write the CPUREC statement before making the ISPF or Job Scheduling Console definition, as they alone 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 19. Defining an Extended Agent Workstation

Chapter 5. Defining Extended Agent Jobs

This chapter explains how to define the Tivoli Workload Scheduler jobs associated with the external jobs that you are interested in running through the extended agents.

To run and monitor a PeopleSoft, R/3, Oracle, or z/OS job with Tivoli Workload Scheduler, the extended agent (or access method) requires a Tivoli Workload Scheduler job definition, where you specify the external job you intend to schedule, the workstation (also defined in Tivoli Workload Scheduler) on which it is supposed to run, and the desired recovery actions. To define the job, you can use one of the following:

- The Tivoli Job Scheduling Console
- The Tivoli Workload Scheduler command line

If you are scheduling in an end-to-end environment, to define the job you must use the Tivoli Workload Scheduler for z/OS ISPF dialogs. In addition, you must create a member in the SCRIPTLIB with a JOBREC statement for the job.

Extended agent jobs are added to job streams and scheduled just like any other job in Tivoli Workload Scheduler and Tivoli Workload Scheduler for z/OS.

Defining Jobs with the Job Scheduling Console

The steps described here for defining an extended agent job are common to all access methods. Also, the access method for R/3 provides supplementary features if you use the alternative steps described in “Using the SAP-Specific Windows of the Job Scheduling Console” on page 100.

For all access methods, follow these steps:

1. In the Actions list of the Job Scheduling Console main window, select **New Job Definition**.
2. Select a Tivoli Workload Scheduler engine and from the resulting list click **Extended Agent Task**. The General page of the Properties - Job Definition window is displayed.

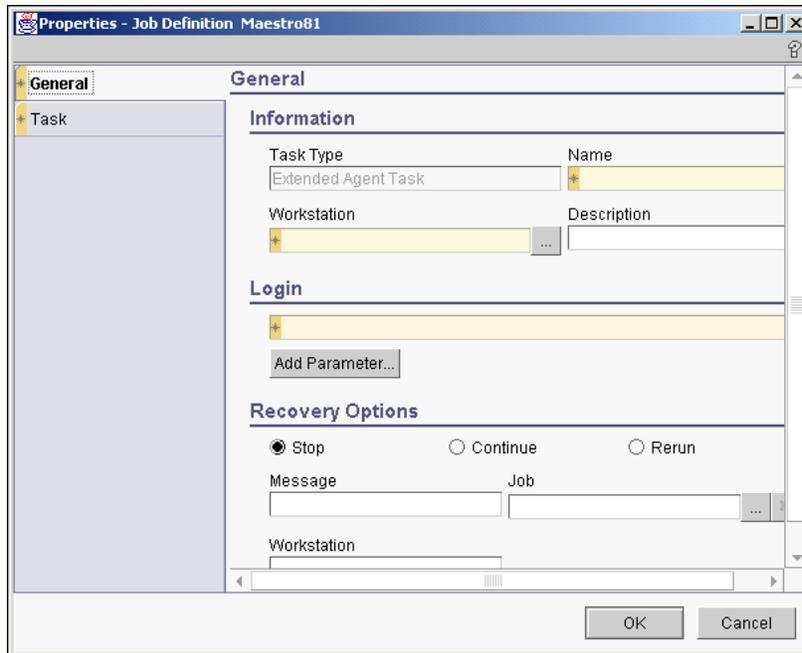


Figure 20. The General Page of the Properties - Job Definition Window.

3. In this page you provide information about the Tivoli Workload Scheduler job that will drive the external job. Complete the fields in this window with the help of the following table.

Table 7. The General Page of the Properties - Job Definition Window (Field Descriptions).

Field	Description
Task Type	This is a read-only field and should state Extended Agent Task .
Name	The name of the Tivoli Workload Scheduler job you are defining. This job will drive the external application job whose name you will provide in the next page of this window. The name can contain up to 8 alphanumeric characters if the database is unexpanded, or 40 if it is expanded. It can include dashes and underscores, but must start with a letter. Embedded spaces are not allowed. It can be the same as the external job name, as long as it is valid in Tivoli Workload Scheduler.
Workstation	The name of the extended agent workstation on which the job is supposed to run. You can click the ellipsis (...) button to display a Find window and select an available workstation.
Description	A text description of the job of up to 64 alphanumeric characters in length. This field is optional.
Login	The user name needed to run the job in Tivoli Workload Scheduler. Normally, this is the value of the LUser parameter, specified during the installation of Tivoli Workload Scheduler for Applications, that is in the options file for the method. This must be a valid user on the Tivoli Workload Scheduler host workstation who must also be able to access the external product. It is required to schedule the job in Tivoli Workload Scheduler, and it will appear in Tivoli Workload Scheduler displays. Optionally, click Add Parameter to add any predefined parameters to the logon ID.

Table 7. The General Page of the Properties - Job Definition Window (Field Descriptions). (continued)

Field	Description
Recovery Options	The automatic response of Tivoli Workload Scheduler if the job abends. The Default is Stop . For information on recovery options see the <i>Tivoli Workload Scheduler Reference Guide</i> .
Message	An optional message text or prompt that is displayed when the job abends. No predefined prompts are accepted here.
Job	The name of a recovery job that starts automatically when the job abends. Click the ellipsis (...) to display a Find window and list existing jobs.
Workstation	The name of the workstation that owns the recovery job.

- Click **Task** to proceed to the next page. The Task page of the window is displayed.

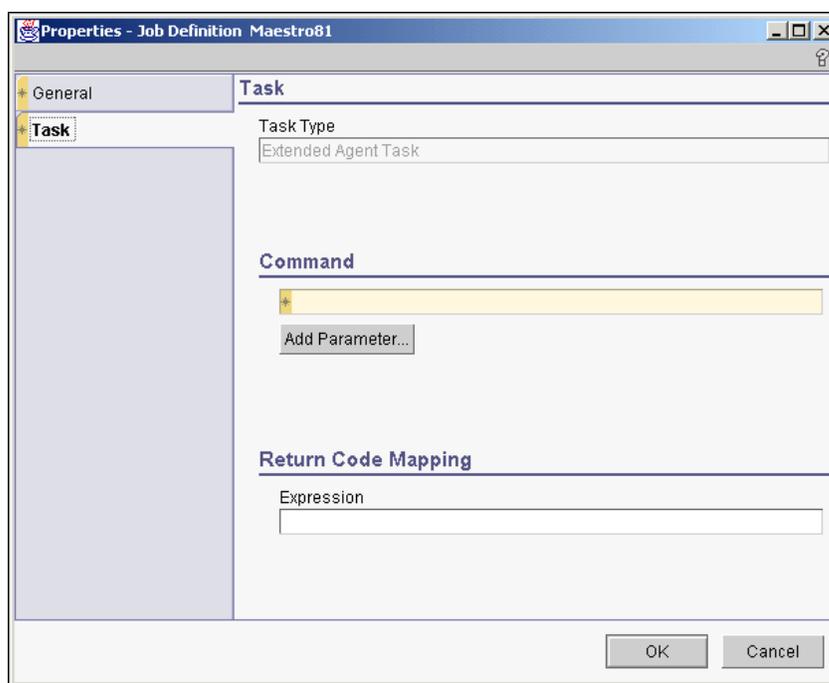


Figure 21. The Task Page of the Properties - Job Definition Window.

- In this page you provide information on the external job (or task) that will be driven by the job defined in the previous page. Note that the task must have been already defined in the external application. Complete the fields in this window with the help of the following table.

Table 8. The Task Page of the Properties - Job Definition Window (Field Descriptions).

Field	Description
Task Type	This is a read-only field and should state Extended Agent Task .

Table 8. The Task Page of the Properties - Job Definition Window (Field Descriptions). (continued)

Field	Description
Command	<p>The specifications for the external job (task). Each external application has a different set of parameters:</p> <ul style="list-style-type: none"> • For Oracle Applications jobs, see “Task String Parameters for Oracle Applications Jobs” on page 47. • For Peoplesoft jobs, see “Task String Parameters for PeopleSoft Jobs” on page 75. • For R/3 jobs, see “Task String Parameters for R/3 Jobs” on page 107. • For z/OS jobs, see “Task Definition Syntax for z/OS Jobs Scheduled with Tivoli Workload Scheduler” on page 149. <p>Write the parameters in a string having the following format <code>-parameter1 value1 -parameter2 value2...-parameterN valueN</code></p> <p>Clicking on Add Parameter... displays a Find Parameters window where you can list and add pre-defined Tivoli Workload Scheduler parameters to the Command field.</p>
Return Code Mapping Expression	<p>A logical expression that defines which return codes cause the task to be successful. You can use comparison and logical operators. For example, the expression <code>RC≥6</code> indicates that a return code greater than or equal to 6 means the task was successful.</p>

6. Click **OK** to save the job definition.

Defining Jobs with the Tivoli Workload Scheduler Command Line

Jobs can also be defined in the **composer** command line of Tivoli Workload Scheduler. The following example documents a Tivoli Workload Scheduler job named `orajob2` that runs in an Oracle Applications extended agent workstation named `ora002`. It logs on to UNIX 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 will run job `recov2` and then continue 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
```

Note that the arguments of `scriptname` differ by application. For reference, see:

- “Task String Parameters for Oracle Applications Jobs” on page 47.
- “Task String Parameters for PeopleSoft Jobs” on page 75.
- “Task String Parameters for R/3 Jobs” on page 107.
- “Task Definition Syntax for z/OS Jobs Scheduled with Tivoli Workload Scheduler” on page 149.

For more information about using the command line to define jobs, refer to the *Tivoli Workload Scheduler Reference Guide*.

Defining Jobs for End-to-end Scheduling

Extended agent jobs scheduled to run in an end-to-end environment cannot be defined through the Job Scheduling Console or the Tivoli Workload Scheduler command line. They 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 the choice of an extended agent workstation denotes their kind. The following example shows the definition of a job named SAPJOB. This is the Tivoli Workload Scheduler for z/OS job that is to drive the execution of an R/3 job (named BAPRINT46B as you will see in the next example). It shows as an extended agent job just 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      Morepreps
cmd  ws       no.   HH.MM.SS                    -IntExt-
'''  R3XA 001   00.00.01  SAPJOB_____          0 0
***** Bottom of data *****
```

Figure 22. Defining an Extended Agent Job for End-to-end Scheduling.

For each job, a member must be created 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 prior to job execution. The job of this example, BAPRINT46B, was previously defined on R/3 and assigned with an ID of 14160001, that was manually transcribed in JOBCMD.

The following example is for a PeopleSoft job. Note that the entire string that follows the JOBCMD keyword must be enclosed by double rather than single quotation marks. This is true only when PeopleSoft jobs are involved, because single quotes are already used in the string.

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

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

- “Task String Parameters for Oracle Applications Jobs” on page 47.
- “Task String Parameters for PeopleSoft Jobs” on page 75.
- “Task String Parameters for R/3 Jobs” on page 107.

- “Task Definition Syntax for z/OS Jobs Scheduled with Tivoli Workload Scheduler” on page 149.

Part 2. Oracle e-Business Suite Access Method

Chapter 6. Introduction

IBM Tivoli Workload Scheduler for Applications for Oracle e-Business Suite gives you the ability to schedule Oracle e-Business Suite jobs using the sophisticated job scheduling features of Tivoli Workload Scheduler.

Throughout this manual, Oracle Applications and Oracle e-Business Suite are used interchangeably to refer to the same product.

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

Features

- Use Tivoli Workload Scheduler's 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 R/3 and PeopleSoft.
- Define inter-dependencies between Oracle Applications jobs and jobs that run on different platforms.
- Customize job execution return codes.

Software Requirements

To install and use IBM Tivoli Workload Scheduler for Applications for Oracle e-Business Suite, you must have:

- One of the following supported operating systems:
 - HP-UX 11.0, 11.i
 - Solaris Operating Environment Versions 7, 8, 9
 - IBM AIX 4.3.3, 5.1
- The following software installed:
 - IBM Tivoli Workload Scheduler 7.0 or higher
 - Oracle Applications 10.7 or Oracle e-Business Suite 11.0 or 11.i
 - For AIX and HP users only, the gcc compiler for the respective platform. The gcc compiler binaries may be retrieved from several sources on the web.

For the latest list of supported platforms, contact a Customer Support representative.

Post-installation Tasks

After you have installed the product as explained in Chapter 2, "Installing IBM Tivoli Workload Scheduler for Applications", on page 5, perform the following steps to complete the installation of the access method for Oracle e-Business Suite:

1. Log on as an Oracle Applications Unix user, who has access to all the files under the Oracle Application Top directory.

2. Change to the installed directory:

```
cd <TWSHome>/methods/OracleTMP
```

Note: If you are running the Oracle Applications demo database, substitute the **APPDEMO.env** file in steps 3 and 4.

3. Modify the Oracle Applications **APPLSYS.env** file as follows:

- a. Append MCM to the APPLSHAR and APPLFULL variables.

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

```
APPLFUL='FND AD AX AK GL RG INV OP MCM'
```

- b. Define and export the shell variable MCM_TOP:

```
MCM_TOP=$APPL_TOP/mcm/integration
```

```
export MCM_TOP
```

- c. Add \$MCM_TOP/bin to the PATH variable by entering the following lines at the bottom of the **APPLSYS.env** file:

```
PATH=$PATH:$MCM_TOP/bin
```

```
export PATH
```

4. Run the **APPLSYS.env** file to set the environment:

```
. APPLSYS.env
```

Note: Make sure that the Oracle Applications logon has permissions to write in the *TWSHome* and **methods** directories.

5. Change the user and group ownership of the installed files to that of your Oracle Application owner.

6. Change directory to /temp/usrxit.

7. For AIX and HP users only:

- Because of the format of MCMxxx.o objects, you must use the gcc compiler for your platform to run the following steps. The gcc compiler binaries may be downloaded from several sources in the internet.

- For AIX users only: Customize the `adrelink` script supplied by Oracle for gcc relinking options:

- a. Find the following line:

```
AIXRIOS) LDFLAGS='-s -H512 -T512 -bhalt:4 -brtl -L$(ORACL ....'
```

- b. Delete

```
-H512 -T512 -bhalt:4 -brtl
```

from it, so that it looks like the following:

```
AIXRIOS) LDFLAGS='-s -L$(ORACL ....'
```

- Ensure that the gcc compiler is configured in the Application Owner environment path. As an alternate option, you can edit the `mcm.mk` file shipped with the product to write the complete gcc path in the line `CC="gcc"`

Note: IBM Tivoli Workload Scheduler for Applications for Oracle e-Business Suite supports only version 2.x of the gcc compiler. Version 3.x is not supported.

8. Run the installation script **mcminstall** from the temp directory. Use the following command:

```
sh mcminstall
```

An Oracle Tivoli screen is displayed.

9. Respond to the requests on the screen according to your Oracle Applications settings and to those of your Tivoli Workload Scheduler installation.

The installation process now attempts to relink the Tivoli Workload Scheduler for Applications executables with the Oracle Applications shared libraries

After the relinking process has completed successfully, you will find the following executables in the \$MCM_TOP/bin directory:

- MCMLJ
- MCMSTAT
- MCMTJ

These binary files are created by the `adrelink` utility for Oracle Applications. This utility will also register the MCM as a module in AOL.

If you encounter problems during relinking, you must proceed to fix them and to run again the relinking step, because this step is responsible for creating the binary files that enable Tivoli Workload Scheduler for Applications to interact with the Oracle Concurrent Manager in submitting and monitoring jobs.

10. Edit the options file. See Chapter 8, “MCMAGENT Options File Reference”, on page 51 and Chapter 3, “Setting Options with the Option Editor”, on page 25 for details.

The installation is now complete. See the IBM Tivoli Workload Scheduler documentation to learn how to start Batchman, to define workstations and jobs, and other Tivoli Workload Scheduler operations.

Interface Software

The interface between Tivoli Workload Scheduler and Oracle e-Business Suite is comprised of the following software:

MCMAGENT

The IBM Tivoli Workload Scheduler for Applications for Oracle e-Business Suite access method.

MCMLJ

Submits a job to Oracle Concurrent Manager (OCM).

MCMSTAT

Monitors the status of the job submitted.

MCMTJ

Terminates the job submitted, on receiving a kill signal from Tivoli Workload Scheduler.

mcm.mk

A makefile utility for linking the above programs with Oracle Applications.

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 Chapter 8, “MCMAGENT Options File Reference”, on page 51 and Chapter 3, “Setting Options with the Option Editor”, on page 25 for details.

<WORKSTATION_NAME>_MCMAGENT.opts

A local options file which contains configuration settings for the extended

agent. One is required for each extended agent workstation that you define. See Chapter 8, "MCMAGENT Options File Reference", on page 51 and Chapter 3, "Setting Options with the Option Editor", on page 25 for details.

Chapter 7. Job Definition Information

This chapter provides information for defining Oracle Applications jobs in Tivoli Workload Scheduler.

Defining Jobs in Oracle Applications

In Oracle Applications, jobs that are to be scheduled with Tivoli Workload Scheduler are defined in the same manner as other Oracle Applications jobs (and can be submitted in OCM). When you are submitting the job in OCM, specify the scheduling and resubmission information. When you are submitting the job through Tivoli Workload Scheduler, Tivoli Workload Scheduler will take care of 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 OCM as well).

Running a Single Job Concurrently

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

Defining Oracle Applications Jobs in Tivoli Workload Scheduler

Tivoli Workload Scheduler job definitions are required for each Oracle Applications job you intend to schedule. Oracle Applications jobs are defined in Tivoli Workload Scheduler like other jobs and include job name, user name, and recovery options. See Chapter 5, "Defining Extended Agent Jobs", on page 35.

Task String Parameters for Oracle Applications Jobs

This section describes the task string parameters that rule the execution of Oracle Applications jobs. You must specify them in the following places when you define their associated Tivoli Workload Scheduler jobs:

- In the **Command** field of the Task page of the Properties - Job Definition Window, if you use the Job Scheduling 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 `JOB_CMD` 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 parameters are described in the following table.

Table 9. Task String Parameters for Oracle Applications Jobs.

Parameter	Description
-user <i>appslogin</i>	The name of a valid Oracle Applications user (Apps login).
-job <i>jobspecs</i>	<p>The attributes of the Oracle Applications job. There are four job attributes that must be specified in the following order:</p> <ul style="list-style-type: none"> • The name of the Applications module used to sign on to Oracle Applications (for example, FND). • A valid responsibility for the Applications module (for example, 'Application Developer'). This parameter must be enclosed in single quotes. • The name of the application that registered the job (for example, P0). • The job name (for example, POXACR). The job name must be uppercase. <p>For example: -job FND 'Application Developer' P0 POXACR</p>
-prn <i>printopts</i>	<p>The two printer options for the job. They must be entered in the following order:</p> <ul style="list-style-type: none"> • The printer name (optional). If omitted, the default printer specified in the options file is used. • The number of copies. <p>For example: -prn ps4 2</p>
<code>[-vn ('value' null :parameter)]</code>	<p>The job's parameters, if any. This string is optional if no parameters are specified for the job in Oracle Applications.</p> <ul style="list-style-type: none"> • <i>n</i> is the parameter number (v1, v2, etc.). • One of the following: <ul style="list-style-type: none"> value An actual value. Enclose <i>value</i> in single quotes if it contains special characters. :parameter The name of an already defined Tivoli Workload Scheduler parameter. See the <i>Tivoli Workload Scheduler Reference Guide</i> for more information about Tivoli Workload Scheduler parameters. <p>The access method will automatically add all the parameters that you do not specify in the definition with a value of null.</p> <p>These are two examples: -v4 'abc' -v15 :p1</p>

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 Applications Job States

When an Oracle Applications job is launched by Tivoli Workload Scheduler, you can monitor its progress with **conman**. The state transitions in Tivoli Workload Scheduler and the corresponding Oracle Applications states are listed below.

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 the *Tivoli Workload Scheduler Reference Manual*.

Chapter 8. MCMAGENT Options File Reference

This section provides detailed reference information about the IBM Tivoli Workload Scheduler for Applications for Oracle e-Business Suite options file.

The Options File

Every installed instance of the Oracle 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 Chapter 3, “Setting Options with the Option Editor”, on page 25) or any common text editor. The naming convention of every options file must be `<WORKSTATION_NAME>_MCMAGENT.opts` and must be located in the `TWSHome/methods` directory. If you do not create one, the agent uses by default the global options file named `MCMAGENT.opts`.

Entry Syntax

The following table lists the options for the MCMAGENT access method.

Table 10. MCMAGENT options.

Option	Description
ORACLE_HOME	The Unix directory where Oracle software is installed.
ORACLE_SID	The Oracle system identifier of the database instance you are connecting to. This corresponds to the value of the Oracle Applications \$ORACLE_SID variable.
ORA_APP_TOPDIR	The Unix directory where Oracle Applications is installed.
FND_TOPDIR	The FND top directory where the Application Object Library resides.
PATH	The setting of the PATH variable, including, the existing path, the path to the bin directory of Oracle Applications, the path to the include directory under the FND Top directory, and others as needed.
ENV	The environment file to execute. Each instance of Oracle Applications has its own environment file (APPLSYS.env or APPDEMO.env). The .env filename given here will be used to set the environment dynamically when Tivoli Workload Scheduler runs jobs on the Oracle Applications x-agent.
USERID	A database user id that has access to all the Modules in Oracle Applications. It must be able to launch, manage and terminate jobs in Oracle Applications.
PASSWORD	The password for the selected userid. To obtain the encrypted value, use the enigma utility shipped with the product. See “Encrypting Oracle Applications User Passwords” on page 52.
PRINTER_NAME	The default printer id.
MIN_TIME	The number of seconds to be added to the previous polling interval after which the job status check has to be carried out.

Table 10. MCMAGENT options. (continued)

Option	Description
MAX_TIME	The number of seconds after which the polling interval will not be incremented.

All the options are required and must be written one per line in the format:

```
OPTION=VALUE
```

with no blanks in between.

Example

File HP113_MCMAGENT.opts is a sample options file:

```
ORACLE_HOME=/general/ora716
ORACLE_SID=DEM0106
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
```

Encrypting Oracle Applications 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.

Part 3. PeopleSoft Access Method

Chapter 9. Introduction

IBM Tivoli Workload Scheduler for Applications for PeopleSoft enables you to execute and monitor PeopleSoft jobs from the Tivoli Workload Scheduler environment. These jobs can be executed as part of a schedule or submitted for ad-hoc job processing. PeopleSoft extended agent jobs can have all of the same dependencies, and recovery options as other Tivoli Workload Scheduler jobs. PeopleSoft jobs must be documented in Tivoli Workload Scheduler to be executed and controlled in the Tivoli Workload Scheduler environment.

Features

- The extended agent for versions 7.0, 7.5 and 8.x of PeopleSoft has been certified by PeopleSoft on Windows NT 4.0 with Service Pack 6 and on Windows 2000.
- PeopleSoft 8.x support.
- 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 R/3 and Oracle.
- Define inter-dependencies between PeopleSoft jobs and jobs that run on different platforms.
- Customize job execution return codes.

Functional Overview

Tivoli Workload Scheduler can launch and monitor jobs in the PeopleSoft process scheduler using a PeopleSoft extended agent workstation. The PeopleSoft extended 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 executes the **psagent** method, passing it information about the job. An options file provides the method with path, executable and other information about the PeopleSoft process scheduler and application server used to launch the job. The extended agent can then access the PeopleSoft process request table and make an entry in the table to launch the job. Job progress and status information are written to the job's standard list file.

PeopleSoft Process Scheduler Jobs

The PeopleSoft process scheduler uses entries in the Process Request table (PSPRCRQST) to drive its work decisions. These entries describe the processes which the Process Scheduler server agent will launch. They also contain the information required to determine the status of the process. The access method for PeopleSoft can insert entries, inquire against the contents and modify the status information within this table (PSPRCRQST).

When documenting a PeopleSoft job in Tivoli Workload Scheduler, the user is actually defining a process request to be inserted in the Process Request table (PSPRCRQST). A process request from Tivoli Workload Scheduler must contain three valid pieces of data:

- A process definition having two parts:
 - Process Type
 - Process Name
- An operator ID (not required on execution on PeopleSoft 8.x)
- A run control ID

All of these items are required for a PeopleSoft job to execute. The access method for PeopleSoft directly updates the PSPRCRQST table for PeopleSoft 7.x (2-tier connection), whereas for PeopleSoft 8.x it exploits the PeopleSoft application server through the interface exposed by the PROCESSREQUEST component interface (3-tier connection).

Note that the process scheduler and the application server run on Unix and on supported Windows FTA workstations, whereas the extended agent runs on Windows NT or Windows 2000 only.

PeopleSoft Job Tracking in Tivoli Workload Scheduler

A job in PeopleSoft 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.

Notes on PeopleSoft Version 8.x Support

PeopleTools 8.x support has some changes that you should know if you plan to use the x-agent method in your environment.

- The `-operator` keyword used in Tivoli Workload Scheduler job specifications, and its corresponding parameter, are currently ignored by the `psagent` method and runtime. As a consequence, jobs will be submitted and will run with the same PS operator ID that is used for connecting with PeopleSoft. You do not have to change your Tivoli Workload Scheduler job definitions if you migrate from PeopleTools 7.x to PeopleTools 8.x. The difference will be that all jobs submitted to a PeopleSoft 8.x x-agent will execute under the same ID specified in the `<CPU_NAME>_psagent.opts` file.
- There is no need to install Database connectivity on FTAs hosting PeopleSoft extended agents because the method currently uses the PeopleSoft 3-tier architecture. You will have to configure at least one PeopleSoft Application Server for the extended agent to work. The application Server must be active in order to successfully submit jobs to the PeopleSoft process scheduler.

- You will not need to load TVPUPSFT.dms into the PeopleSoft 8.x Database when working in 3-tier mode.
- For PeopleSoft 8.1x, the method requires the installation of PeopleSoft External Api support on the FTA that is hosting the extended agent method. The installation directory of the External API is used during the extended agent installation and configuration processes when you are required to specify the PeopleSoft home directory or the COBOL root directory. To install the external API use the setup.exe executable located in your PeopleSoft 8.1x fileserver directory named: <PS_HOME>\setup\PsExtAPI. Installation of PeopleSoft External API needs to be performed before the installation of Tivoli Workload Scheduler for Applications. Verify your External API installation as directed by the instructions in PeopleSoft: using the jnittest.exe tool located in the PS External API installation directory. You will need to reboot your workstation before you can use thePsExtAPI installation successfully. If you fail to reboot you will see a popup window from TVPUPSFT.exe saying:
"The application failed to initialize properly (0xc0000005). Click on OK to terminate the application"

Chapter 10. Implementation Planning

PeopleSoft Version 7.5 or Earlier

This section describes implementation planning for the extended agent with PeopleSoft 7.5 or earlier.

Host Requirements

The workstation hosting the extended agent has the following requirements:

- It must be a Tivoli Workload Scheduler fault-tolerant agent (FTA) or standard agent (SA) running on a supported Windows version.
- It must have network access to the PeopleSoft database.
- It must be capable of being a PeopleSoft workstation.
- It must have access to the PeopleSoft and COBOL support libraries. COBOL support must be compatible with the compiler used for your NT process scheduler environment.

Software Requirements

IBM Tivoli Workload Scheduler for Applications for PeopleSoft has the following software requirements:

- Windows NT 4.0 with Service Pack 6 or Windows 2000 with service pack 2.
- IBM Tivoli Workload Scheduler Version 7.0 or later
- PeopleSoft version 7.0, 7.5
- PeopleSoft COBOL Compiler

Notes:

1. The extended agent for versions 7.0 and 7.5 of PeopleSoft is available with MicroFocus NetExpress 2.1 and MicroFocus NetExpress 3.1.
2. The extended agent for PeopleSoft version 7.0 was also compiled on Micro Focus COBOL workbench Compiler version 4.1
3. For Peoplesoft versions 7.x, you are required to recompile your Windows Peoplesoft process scheduler if you will be using it for jobs managed with Tivoli Workload Scheduler.

Windows User Setup

- Create a local or domain User ID with authority to access both PeopleSoft and Tivoli Workload Scheduler software or set up the existing user to allow access to the PeopleSoft software and Tivoli Workload Scheduler. See “User Setup for Windows” on page 63 for details.
- Log on as the new user.
- Connect to the appropriate drives for PeopleSoft. Be sure to select **Reconnect at Logon**. The drive letter should be the same as the one you have specified as PSDRIVE in `psagent.opts`.
- Perform a PeopleSoft setup. Follow the instructions supplied with your PeopleSoft software.

Verifying the PeopleSoft Environment

Make sure that your PeopleSoft environment is properly configured and that PeopleSoft jobs can be executed successfully by the process scheduler.

Notes:

1. Verify database connectivity to PeopleSoft database using the Windows User ID created earlier. The connection can be checked by running a database command using any ODBC utility.
2. Verify access to the Peoplesoft file server or availability of local copy of executables.

Creating a Batch Processing ID in PeopleSoft

You should create an operator ID in PeopleSoft dedicated to batch scheduling. All the jobs submitted by Tivoli Workload Scheduler should use this operator ID.

Installed Files

The following files are installed in Tivoli Workload Scheduler and PeopleSoft directories during the installation of Tivoli Workload Scheduler for Applications:

Destination	Files	Owner
<i>TWSHome</i> \methods\PeopleSoft	<ul style="list-style-type: none">• TVPUPSFT7xx.dms, where xx specifies the release of PeopleSoft Version 7.• PTCUDCOP.cbl	Tivoli Workload Scheduler User
<i>TWSHome</i> \methods	<ul style="list-style-type: none">• psagent.exe• enigma.exe	Tivoli Workload Scheduler User
<PSDRIVE_PSVER>\cblbin	<ul style="list-style-type: none">• TVPUPSFT.exe• TVPUPSFT.gnt	Owner of PTPUPRCS.exe

Note: *TWSHome* stands for the name of the directory where you installed Tivoli Workload Scheduler.

PeopleSoft Version 8.x

This section describes implementation planning for the extended agent with PeopleSoft version 8.x.

Host Requirements

The workstation hosting the extended agent must be a Tivoli Workload Scheduler fault-tolerant or standard agent in a supported Windows platform.

If your site has multiple process servers for a single PeopleSoft database, you can configure one extended agent workstation (using multiple options files) to process jobs on multiple process servers.

Software Requirements

IBM Tivoli Workload Scheduler for Applications for PeopleSoft has the following software requirements:

- Windows NT 4.0 with Service Pack 6 or Windows 2000 with service pack 2.
- IBM Tivoli Workload Scheduler Version 7.0 or higher.

- PeopleSoft version 8.1x or 8.4x.
- For PeopleSoft version 8.1x: PeopleTools External APIs version 8.1x.
- For PeopleSoft version 8.4x: access to the PeopleSoft File Server 8.4x or availability of a local copy of executables.

Creating a Batch Processing ID in PeopleSoft

You should create an operator ID in PeopleSoft dedicated to batch scheduling. The 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.

Installed Files

The following files are installed in IBM Tivoli Workload Scheduler and PeopleSoft directories during the installation of IBM Tivoli Workload Scheduler for Applications for PeopleSoft:

Destination	Files	Owner
<i>TWShome</i> \methods	<ul style="list-style-type: none"> • psagent.exe • enigma.exe 	Tivoli Workload Scheduler User
<i>PSAPI_dir</i>	TVPUPSFT.exe	Tivoli Workload Scheduler User

Note: *TWShome* stands for the name of the directory where you installed Tivoli Workload Scheduler.

Chapter 11. Post-installation Tasks

This chapter describes important tasks that must follow installation of the access method.

User Setup for Windows

When a PeopleSoft client is used to submit jobs with IBM Tivoli Workload Scheduler for Applications for PeopleSoft in the three tier environment using a share on a PeopleSoft file server, the streamlogon specified on the job's submission must be "known" on both the submitting client and on the PeopleSoft fileserver. In the sense that it must be either:

- A domain user
- A local user defined on both the PeopleSoft fileserver and on the PeopleSoft client with the same name and same password. The user must then be defined in the Tivoli Workload Scheduler database.

Not meeting this condition will produce the following symptoms:

- Jobs will abend
- Two operating system error messages will be sent to the job's stdlist, saying:
Logon failure: unknown user name or bad password.
procwork.cpp(328): The pipe has ended.

The user must also be part of Administrator's Group on the PeopleSoft client. Failure to meet this condition will produce the following symptoms:

- Jobs will go in INTRO state (and a Plus "+" sign will be added to indicate an active job)
- A popup window will be shown on the Tivoli Workload Scheduler workstation hosting the extended agent, with a window title psagent.exe - Application Error, saying:
The application failed to initialize properly (0xc0000005).
Click on OK to terminate the application.
- After clicking on the OK button the jobs will go in the abend state, and the stdlist of the job will only show the banner of the JOBLNCH program.

Creating the Options File

The PeopleSoft extended agent options files specify, among other things, with which database user the method connects to the PeopleSoft database. They are also used to specify the polling rate to the PeopleSoft module for checking job status.

This file must reside on the host computer for the extended agent in the *TWShome\methods* directory.

To create an options file you can use either the Option Editor available with this product, or any text editor of your liking. See Chapter 3, "Setting Options with the Option Editor", on page 25 for details on creating or editing options files with the Options Editor and "Manually Editing Options Files" on page 66 for examples of options files for this access method.

The following table describes the options for the psagent access method:

Table 11. Psagent Options.

Option	Description
LJuser	The ID of the Windows user who launches the IBM Tivoli Workload Scheduler jobs. This Windows user must be a valid Tivoli Workload Scheduler user on the host workstation.
CFuser	The ID of the Windows user who executes a check file. The default value is to match the Launch Job User.
CheckInterval	The time interval between status checks when monitoring a process request. The default value is 2 minutes. Use the CheckIntervalUnits option to change from minutes to seconds.
CheckIntervalUnits	Specifies that the Status Check Interval is in seconds (value=1) rather than minutes (value=60).
PSXA_TRACE	A value of 1 enables trace execution. A value of 0 disables it.
PSDRIVE	The Windows drive letter for the PeopleSoft files (for example, N:).
PSVER	The home directory to the PeopleSoft files (for example, \PT7.5 or \PT8) for all PeopleSoft versions but 8.1x. In this case, it is the directory where the TVPUPSFT.exe file is installed (for example, \PT817\bin\client\winx86_extapi)
COBROOT	<p>For PeopleSoft 7.x: the path to the base directory for the Micro Focus COBOL libraries (for example, N:\PT7.5).</p> <p>For PeopleSoft 8.x: the path to the base directory for the PeopleSoft External APIs. It is the same as the PeopleSoft Home Directory.</p>

Table 11. Psagent Options. (continued)

Option	Description
CONNECT	<p>A string that contains the following entries:</p> <p>Connection type The PeopleSoft RDBMS system type. Choose one of the following:</p> <ul style="list-style-type: none"> • APPSERV (PeopleSoft 8.x only) • ORACLE • MICROSOFT (Microsoft SQL Server) • DB2® (DB2 for MVS™) • DB2UNIX • SQLBASE • DB2ODBC • DB2400 • INFORMIX • SYBASE • DB2MDI <p>Database server name The Database Server Name. This is for Informix™ and Sybase only. Not used for PeopleSoft Version 8.x.</p> <p>Server name The name (IP address) of the server specified as Connection type.</p> <p>For APPSERV, this entry must be either the IP address:TCP port or the IP name.</p> <p>You can specify multiple entries as long as they are separated by commas.</p> <p>Operator ID The operator ID used to connect to the PeopleSoft database.</p> <ul style="list-style-type: none"> • For PeopleSoft Version 7.x, all jobs will not be run as this ID, but as the ID specified in the Tivoli Workload Scheduler job definition. • For PeopleSoft Version 8.x, all jobs will run as this ID. <p>Operator password The encrypted password for the operator ID. If you use a common editor to write the options file, you must first use the Enigma tool to encrypt it. If you use the Option Editor, it does it for you.</p> <p>Default process server The default process server that executes jobs in PeopleSoft.</p> <p>The entries are separated by slashes (/), so that the string is in the form <code>item1/item2/.../item6</code> and must be listed in the sequence displayed in this explanation.</p> <p>All entries are mandatory with the exception of Database server name.</p>

Table 11. *Psagent Options. (continued)*

Option	Description
TWSXA_PT84_JVMPATH	The path to the Java Virtual Machine. This option is available for use with PeopleTools Version 8.4x only.
UTF8cmdline	If set to 0, it means that the extended agent method expects Tivoli Workload Scheduler to send the data necessary for launching the job in the local codepage. Otherwise, it expects the data in UTF8 format. The default value is 1 (for example, UTF8 format).
TWSXA_CP	The codepage that the extended agent method uses while communicating to the external application. If the option is absent or has an invalid value, the local codepage will be used.
TWSMETH_CP	The codepage that the extended agent method uses to generate the standard output. If the option is absent or has an invalid value, the local codepage will be used.
TWSMETH_LANG	The language that the extended agent method uses for the standard output (that is, the specified message catalog will be opened). If the option is absent or has an invalid value, the local language will be used.
TWSXA_LANG	Not supported for the PeopleSoft extended agent.

Option names are case sensitive and must be written as shown in the left column of table 11. Before you use a manually-created options file, check that all the option names are written correctly. Non-complying options will be ignored.

Manually Editing Options Files

Beside the Options Editor, you can also use any common editor to manually create or change an options file.

Below are two example options files. These examples are meant to help you determine your specific site requirements. Your options file may differ from the examples below.

Remember to save the file in the *TWSHome\methods* directory

Example for PeopleSoft 7.x

```
LJuser=TwsUsr
CFuser=TwsUsr
PSDRIVE=N:
PSVER=\PS750
COBROOT=N:\PS750
CONNECT=ORACLE/HR75/PSUSA/#oqo1a4071xp5refgbAreV/PSNT
CheckInterval=15
CheckIntervalUnits=1
PSXA_TRACE=0
```

Example for PeopleSoft 8.1x

```
LJuser=TwsUsr
CFuser=TwsUsr
CheckInterval=15
PSDRIVE=D:
PSVER=\pt81x\bin\client\winx86_extapi
```

```
COBROOT=D:\pt81x\bin\client\winx86_extapi
CONNECT=AppServ/PSOFT81X:9000/PSDEV/jp5183n1We7wh5jxk46qeV/PSUNX
CheckIntervalUnits=1
PSXA_TRACE=1
```

Connecting to More than One PeopleSoft Database

It may be necessary for the **psagent** method to connect to more than one PeopleSoft database. For example, a single installation of PeopleSoft may have a TEST, DEMO, and PRODUCTION environment, each with a separate database. This requires that the **psagent** method use a separate connect string for each database.

In order to support this, you can setup multiple PeopleSoft extended agent workstations that connect to the same method but use different option 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 jupiter would have the following options file:

```
jupiter_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 database.

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

Setting Up the PeopleSoft Environment for Version 7.5 or earlier

The following section explains how to set up the PeopleSoft environment if you will use the extended agent with PeopleSoft version 7.5 or earlier.

Customizing the Process Server

Regardless of the type of operating system where you will be scheduling Tivoli Workload Scheduler jobs, you are required to customize the corresponding process server executable using the COBOL source PTCUDCOP.cbl provided by IBM.

You must replace the PTCUDCOP.cbl file provided by PeopleSoft with the equivalent file provided by IBM and recompile your process scheduler.

Complete directions on how to recompile your process scheduler may be found in PeopleSoft installation manuals. When receiving patches from PeopleSoft, you may have to repeat the process server customization process, if the PTCUDCOP.cbl file is overwritten during patch installation.

Following are the explanations of how to run the customization steps on Windows and on UNIX.

Customizing on Windows

If the PeopleSoft process scheduler runs on Windows, refer to the section "Setting Up the Batch Environment on Windows NT/2000" in the PeopleSoft installation manual for up to date information. You should have installed a COBOL compiler on your batch server and you should compile from that location. The computer doing the compilation must have a COBOL compiler installed and must have write access to the file server.

To build the process server on a Windows computer:

1. Open a DOS command prompt window.
2. Set up two environment variables, PS_HOME and COBROOT, on the computer from which you plan to compile COBOL. Use these commands:

```
set PS_HOME=n:\PS750
set COBROOT=c:\netexpress\base
```

3. Make sure the compilation directory c:\temp\compile exists; it may already exist if you have run CBLBLD.BAT. If it does exist, remove any files residing there, just as a safeguard. If it doesn't exist, you need to create it with the following command:

```
mkdir c:\temp\compile
```

Note: Make sure you have write permission to the compilation directory as the compilation process will take place in that location.

4. Change to the <PS_HOME>\setup directory.
5. Execute the following command to copy all the COBOL source files from the <PS_HOME> directory to the compilation directory:

```
cblsrc <source directory> <compilation directory>
```

where <source directory> is the drive and directory where the source resides (it should be the same as <PS_HOME>), and <compilation directory> is the drive and directory to which the source files will be copied.

For example, the following command will take the COBOL source from <PS_HOME> and copy all the necessary files to the location where the compilation process will take place.

```
cblsrc <PS_HOME> c:\temp\compile
```

6. Copy the file PTCUDCOP.cbl provided by IBM to the compilation directory with the following command:

```
copy <TWS_HOME>\PeopleSoft\PTCUDCOP.cbl c:\temp\compile
```

7. After CBLSRC completes, change the current directory to the compilation directory with the following command:

```
cd c:\temp\compile
```

8. Compile with the following command:

```
CBLMAKE.BAT PTPUPRCS EXE LIST
```

If there are compilation problems, review the PTPUPRCS.lis file for details and retry the command after addressing the problems.

9. After CBLMAKE.BAT completes, copy the EXE, GNT, or INT files to the <PS_HOME>\cblbin directory with the following commands:

```
copy PTPUPRCS.exe <PS_HOME>\cblbin
```

```
copy PTPUPRCS.gnt <PS_HOME>\cblbin
```

Customizing on UNIX

If the PeopleSoft process scheduler runs on UNIX, refer to the section "Setting Up the Batch Environment on UNIX" in the PeopleSoft installation manual for up to date information. You should have installed a COBOL compiler on your batch server and you should be carrying out the compile from that location. The computer doing the compilation must have a COBOL compiler installed and must have write access to the <PS_HOME>/cblbin directory.

To build the process server on a UNIX machine:

1. Log onto the UNIX server.

2. Change the current directory to the PeopleSoft home directory.
3. Run the psconfig.sh script.
4. Change directories to the \$PS_HOME/src/cbl directory.
5. Rename the PTCUDCOP.cbl to PTCUDCOP.old using the following command:

```
mv PTCUDCOP.cbl PTCUDCOP.old
```
6. Copy PTCUDCOP.cbl from the extended agent (*twshome*/PeopleSoft) using FTP.
7. Execute the following commands:

```
cd ../../install
pscb1.mak PTPUPRCS LIST
```
8. After the script has completed, check the destination directories for the newly created files in

```
<PS_HOME>/src/cbl/int
<PS_HOME>/src/cbl/1st
<PS_HOME>/cblbin
```

They should have a length greater than zero as well as a current date and time stamp. If there are problems with the compilation review the PTPUPRCS.lis file for errors, correct them and retry the compilation step.

Setting Up the Client

To run the PeopleSoft access method for a PeopleTools 7.x environment, you have to run in a two-tier PeopleSoft Environment, using a workstation with proper connectivity for your database.

You have to install the client software for your database of choice on the workstation hosting the extended agent and you have to configure it for proper ODBC communications. Test your connection using the ODBC testing tools provided with your client software (for example ODBC 32bit test for Oracle).

You also need access to the PeopleSoft Fileserver. Map the fileserver share to a network drive on your Windows workstation.

For a workstation installation for your PeopleSoft 7.x environment:

1. Access \\<fileserver>\<share> as shared drive <d>: on your computer.
2. Set PS_HOME to the proper directory on the new shared drive with this comand:

```
set PS_HOME=<d>:\PS750
```
3. Run:

```
%PS_HOME%\bin\CLIENT\WINX86\pscfg.exe
```

A configuration window, like that shown in Figure 23 on page 70, is displayed.

4. Select the **Client Setup** tab and fill the required values ensuring that you select the **Install PeopleSoft ODBC Driver** check box. If you run SQL Server, you must also select the **Install ODBC Driver Manager 3.0** check box.

Note: The Install PeopleSoft ODBC Driver flag installs a "user" data source name. If you need to access the data source from another user profile you will need either to create a corresponding System data source, using "Control Panel"-> "Data Source (ODBC)" customization on your WinNT/2000 machine, or to rerun the customization step for all the

other users that you want to use to submit Tivoli Workload Scheduler extended agent jobs.

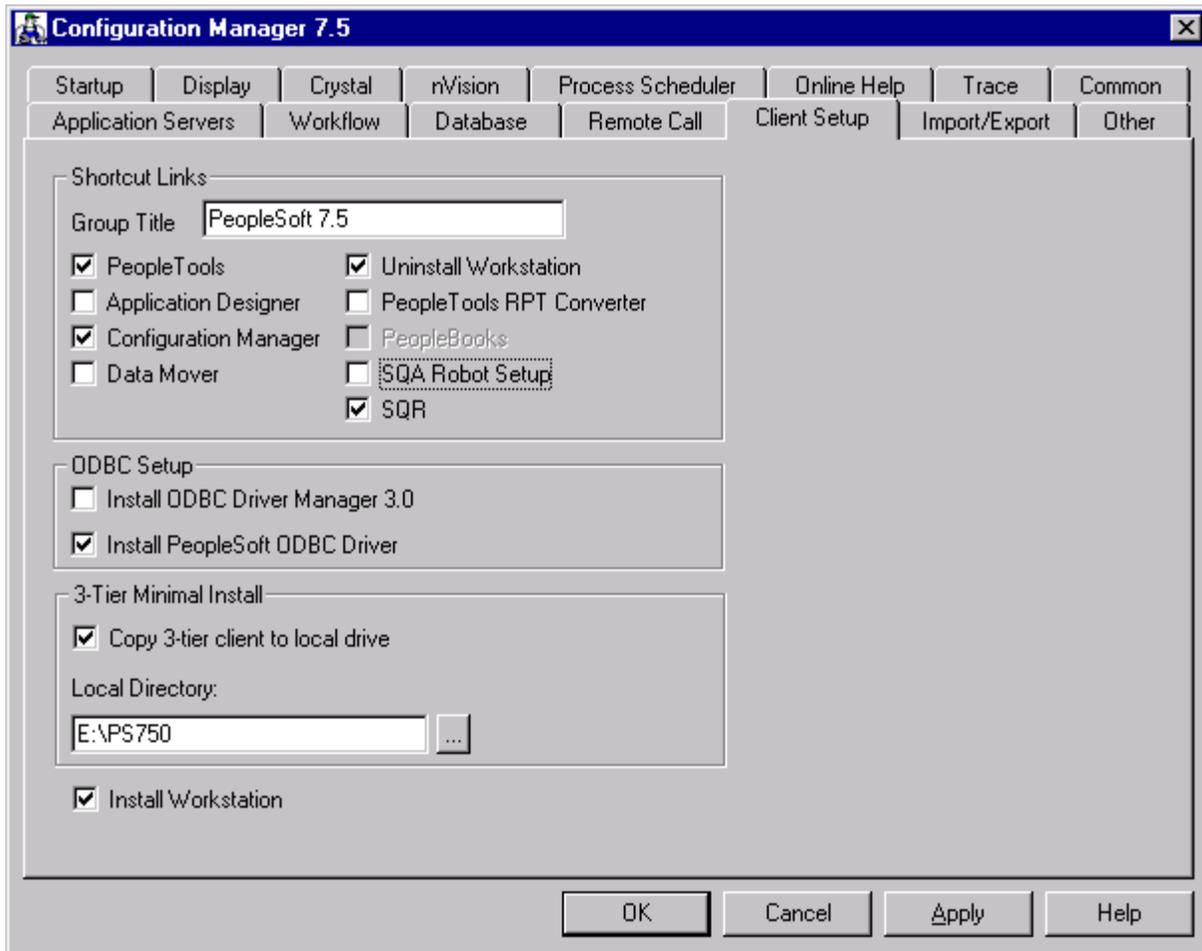


Figure 23. Configuration Manager 7.5 Window.

After you are finished, you should have in your "Start/Programs" menu a new item labelled PeopleSoft 7.5.

Running the Data Mover Script

After the installation sequence is complete the PeopleSoft Data Mover is used to moved the stored procedures for the PeopleSoft extended agent into place. For more information on the Data Mover script consult your Peoplesoft documentation.

1. Click on the DataMover icon in the PeopleSoft group. The Peoplesoft Signon window is displayed.

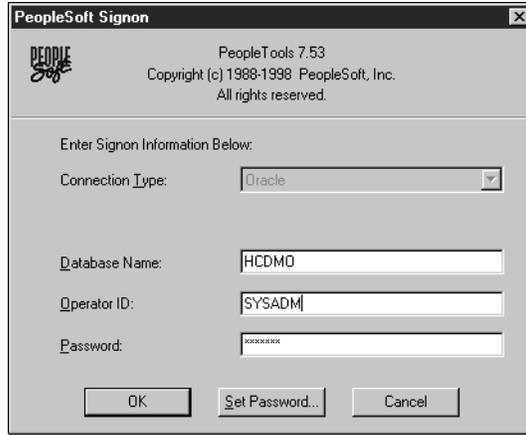


Figure 24. PeopleSoft Signon Window.

2. Select the appropriate database, and click **OK**. The PeopleSoft logon window is displayed.
3. Fill in the logon fields, and click **OK**. The Data Mover window is displayed.
4. From the **File** menu on the Data Mover window, select the **Open** command. Open the following file in the Data Mover window:
TwShome\PeopleSoft\TVPUPSFT.dms
5. Once you have loaded the scripts in the Data Mover window, run the scripts. The Data Mover window is separated into two parts. The upper section shows the Data Mover script statements. The lower part shows the status of the script statements as they execute.
6. Verify that the script runs successfully.

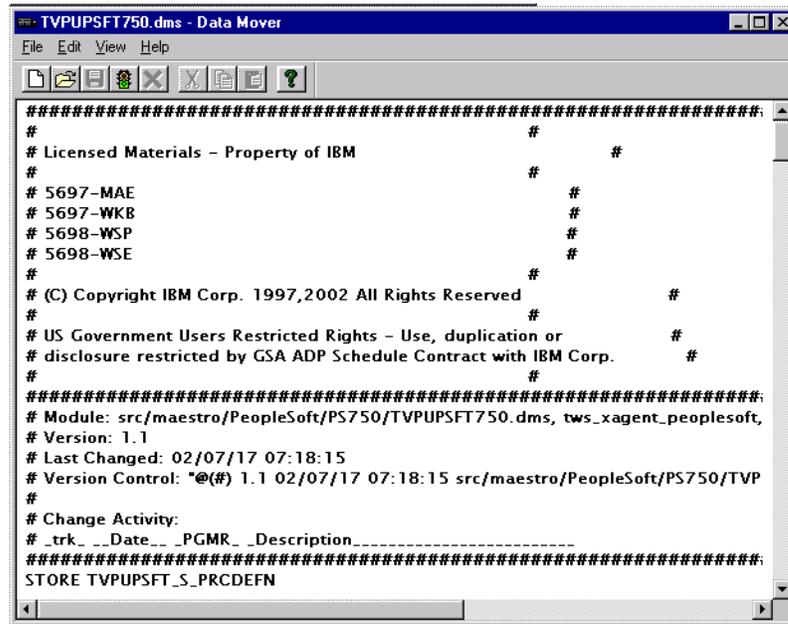


Figure 25. DataMover Window

If statements fail, check the startup window values for correct Database Name, Login ID, and Password, then repeat this process. If the Data Mover script continues to fail, see your DBA to determine if there is a problem in your database.

Note: The Data Mover script must be run on each database the extended agent for PeopleSoft will use. To run the scripts on another database, change the database name and/or RDBMS in the login sequence.

Setting Up the PeopleSoft Environment for Version 8.1x

The following section explains how to set up the PeopleSoft environment if you will use the extended agent with PeopleSoft version 8.1x.

Customizing the Application Server

The access method supporting PeopleTools 8.1x can only work in conjunction with a PeopleSoft 3-tier architecture. This implies that you need to deploy a PeopleSoft 3-tier architecture according with PeopleSoft instructions. While requiring a complete deployment of PeopleSoft architecture with at least one PeopleSoft application server and one process server, this will reduce software requirements and dependencies on your extended agent workstation. There is no need for installing database connectivity on the workstation hosting the extended agent. Moreover, there are no customization requirements for the process server, as it was with PeopleTools version 7.x, and there will be no need for running additional data mover scripts for database customization. The application server must be active in order to successfully submit jobs to the PeopleSoft process scheduler.

To properly configure the access method, you will need to obtain the following information from your PeopleSoft administrator:

- Hostname (or IP address) of the application server
- Application server Jolt port (typically 9000)
- Process server name

The operator ID must be granted authority to use the Component Interface in the PeopleTools environment.

Setting Up the Client

To run the extended agent with a PeopleTools 8.1x environment, you have to install PeopleSoft External API support on the Tivoli Workload Scheduler workstation hosting the extended agent.

Installation of PeopleSoft External API must be performed before the installation of Tivoli Workload Scheduler for Applications for PeopleSoft.

The PeopleSoft installation manuals contain information about this installation step. Refer to the proper PeopleSoft documentation for up to date instructions, depending on your actual PeopleSoft version. For PeopleTools 8.17 you will have to invoke the setup.exe executable located in your PeopleSoft 8 fileservers directory named %PS_HOME%\setup\PSExtAPI.

Make note of the directory where you install the External API, because you will need it during installation of the extended agent and during configuration when you are prompted for the PSHome directory, or when you are asked for the Cobol root directory.

Before using the External API, verify your installation as directed by PeopleSoft instructions, using the `jnitest.exe` tool located in the External API directory. The `jnitest.exe` tool verifies that you have the Microsoft Java Virtual Machine for Java 3.2 version 5.00.3167 or higher. If `jnitest.exe` finds an acceptable version, the Microsoft Java Virtual Machine can be loaded. Otherwise, you may have difficulties at loading it properly. The External API will not function unless you have the right version of the Microsoft Java Virtual Machine.

You will also need to reboot the workstation before you can use the PeopleSoft External API installation successfully. If you fail to do this, whenever a PeopleSoft job is submitted through ITWS, you will be shown a popup window from `TVPUPSFT.exe` saying:

"The application failed to initialize properly (0xc0000005).
Click on OK to terminate the application."

Setting Up the PeopleSoft Environment for Version 8.4x

For PeopleTools Version 8.4x there is no longer a separate installation of PeopleSoft External APIs, that are delivered as part of the standard Windows client; they can be accessed via the PeopleSoft file server under the directory named `PS_HOME\bin\client\winx86`. Additionally, in accordance with PeopleSoft Resolution ID 704309, you must:

- Be able to access the directory named `PS_HOME\class` containing file `psjoa.jar`.
- Have access to a valid Java Virtual Machine (JVM).

Note: The JVM provided with this product is not valid for this purpose. You must use a JVM supported by PeopleSoft.

Method `psagent.exe` will take care of defining/updating environment variables `CLASSPATH` and `PS_HOME` if they are not already defined in the environment where the job is launched. If your system path environment already contains a valid access to `jvm.dll`, then you do not need additional environment setup. If not, you must specify a new option in the `psagent.opts` file with a text editor, so that `psagent.exe` can update the path to where the JVM is located. To do this, add to the options file the following line:

```
TWSXA_PT84_JVMPATH=<path to jvm.dll>
```

Valid JVMs are already provided by PeopleSoft with the application servers that support the PIA architecture, like "BEA WebLogic Server" (in subdirectory `jdk131\jre\bin\classic` or `jdk131\jre\bin\hotspot`) or "IBM WebSphere Application Server" (in subdirectory `Java13\jre\bin\classic`). For instance, you can update the options file with one of the following:

- `TWSXA_PT84_JVMPATH=D:\Program Files\IBM\Java13\jre\bin\classic`
- `TWSXA_PT84_JVMPATH=D:\bea\jdk131\jre\bin\classic`
- `TWSXA_PT84_JVMPATH=D:\bea\jdk131\jre\bin\hotspot`

If you do not have this keyword or if you do not have a valid system path to `jvm.dll`, you will get one of these messages:

- `AWS140020078E` Exception caught in file `TVPUPSFT.cpp(#)` during function `Main`
- `AWS140020076E` The problem encountered may be related to the environment settings: please verify the following:
 - 1) Your path contains a valid Java Virtual Machine directory or options file contains `TWSXA_PT84_JVMPATH`
 - 2) Your `CLASSPATH` environment variable contains a valid path to `psjoa.jar`
 - 3) Your target Application Server is up and running
 - 4) you have defined `PS_HOME` environment variable

Chapter 12. Defining PeopleSoft Jobs

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

Creating Peoplesoft Job Definitions in Tivoli Workload Scheduler

Tivoli Workload Scheduler job definitions are required for each Peoplesoft job you intend to control. A Tivoli Workload Scheduler job is associated to an already defined Peoplesoft job and its definition includes:

- Job name
- Owner name
- Script file specifications
- Recovery options

See Chapter 5, “Defining Extended Agent Jobs”, on page 35 for reference.

Task String Parameters for PeopleSoft Jobs

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

- In the **Command** field of the Task page of the Properties - Job Definition Window, if you use the Job Scheduling 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. Note that in this case the entire string following the `JOBCMD` keyword must be enclosed within double quotes.

The following table lists the parameters.

Table 12. Task String Parameters for PeopleSoft Jobs.

Parameter	Description	PeopleTools Version
-operator	The PeopleSoft operator ID who launches the job.	7.x, 8.1x
-process	The process name for the PeopleSoft job.	7.x, 8.1x, 8.4x
-type	The process type for the PeopleSoft job. This entry must be enclosed in single quotes.	7.x, 8.1x, 8.4x
-runcontrol	The runcontrol ID for the PeopleSoft job.	7.x, 8.1x, 8.4x
-outputdest	The destination of the PeopleSoft job output.	8.1x, 8.4x

Table 12. Task String Parameters for PeopleSoft Jobs. (continued)

Parameter	Description	PeopleTools Version
-outputtype	The output type of the PeopleSoft job. Valid values are: 0 (Any) 1 (None) 2 (File) 3 (Printer) 4 (Window) 5 (Email) 6 (Web) 7 (default)	8.1x, 8.4x
-outputformat	The output format of the PeopleSoft job. Valid values are: None CSV DOC Default HP HTM LP OTHER PDF PS RPT RTF SPF TXT WKS XLS	8.1x, 8.4x
-runlocationdescr	The PeopleSoft process scheduler in charge of processing the PeopleSoft job.	8.1x, 8.4x
-foldername	The name of the report folder used for this job. The folder must have been already created using PeopleSoft Report Manager.	8.4x

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

The following is an example of an entire task string specification for a Peoplesoft 8.1x job:

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

Setting up Opens Dependencies 7.x Only

You can set up Tivoli Workload Scheduler opens dependencies on PeopleSoft jobs. Tivoli Workload Scheduler *opens dependencies* are mapped to an SQL “select count” command with a “where” clause on the PeopleSoft database. This SQL basically checks that values you select exist in the database. If the specified entries do exist

in the database, the count will return with a value and the job will execute. If the specified entries do not exist in the database, the count will return with a 0 and the job will not be allowed to execute. A result of zero is false; any other value is true.

Note: When creating the opens dependency in the Job Scheduling Console, the *table* is listed under the file text box, and the *column* and *data* are listed as qualifiers in the qualifiers text box

For example:

```
opens /PSSERVERSTAT
```

results in the following SQL command:

```
select count(*) from PSSERVERSTAT;
```

This returns a count of the rows in PSSERVERSTAT. A result of zero is false; any other value is true.

Another example is:

```
opens /PSSERVERSTAT ("SERVERNAME = 'PSNT' AND SERVERSTATUS = 3")
```

This results in the following SQL command:

```
select count(*) from PSSERVERSTAT where SERVERNAME = 'PSNT' AND SERVERSTATUS = 3;
```

This returns a count of the rows in PSSERVERSTAT for the server PSNT where the status is 3 (running).

To set up an opens dependency on a job:

1. Edit the job stream where you want to add a opens dependency in the command line mode.
2. Add the following command string to the job:

```
cpu#job opens /table("column = 'data'")
```

where:

table Specifies the name of the PeopleSoft table you want to open.

column Specifies the name of the column you are looking for.

data Specifies the data within that column that matches your search.

3. Save the changes you have made and close the job stream.

Note: The table data that you are searching for must be enclosed in double quotes.

Example Opens Dependencies

The following job streams have examples of opens dependencies.

Here is a first example:

```
SCHEDULE PS_CPU#TEST01
  ON EVERYDAY
:
PS_CPU#XRFWIN1
PS_CPU#XRFAPF1
PS_CPU#XRFNP1   FOLLOWS XRFAPF1
PS_CPU#SYSAUDT1 FOLLOWS XRFWIN1
PS_CPU#PTPTEDT1 OPENS /PSPRCSRQST ("PRCSTYPE = 'SQR Report' AND RUNSTATUS = 9")
```

```

PS_CPU#XRFILD1  FOLLOWS XRFNP1,PTPTEDT1
PS_CPU#DDDAUDT1 FOLLOWS PTPTEDT1,TEST02.@
PS_CPU#TLS7011  FOLLOWS TEST02. @,DDDAUDT1
END

```

Here follows another example:

```

SCHEDULE PS_CPU#TEST02
  ON EVERYDAY
  AT 1330
:
PS_CPU#XRFWIN2  OPENS /PSSERVERSTAT ("SERVERNAME = 'PSNT' AND SERVERSTATUS = 3")
PS_CPU#XRFAPF2  FOLLOWS TEST01.XRFAPF1
PS_CPU#XRFNP2   FOLLOWS XRFAPF2
PS_CPU#SYSAUDT2 FOLLOWS XRFWIN2
PS_CPU#PTPTEDT2 OPENS /PSPRCSRQST ("PRCSTYPE = 'SQR Report' AND RUNSTATUS = 9
AND PRCNAME = 'DDDAUDIT'")
PS_CPU#XRFILD2  FOLLOWS XRFNP2,PTPTEDT2
PS_CPU#DDDAUDT2
END

```

IBM Tivoli Workload Scheduler and PeopleSoft Job Status Mapping

When a PeopleSoft job is launched by Tivoli Workload Scheduler, you can monitor its progress with the Job Scheduling Console. The state transitions in IBM Tivoli Workload Scheduler and the corresponding PeopleSoft states are listed below.

PeopleSoft Status	IBM Tivoli Workload Scheduler Status	Description
CANCEL	CANCEL/ABEND	Either a PeopleSoft operator or the method has marked the process request for cancellation. The method will do this in response to a TJ task.
DELETE	CANCEL	A PeopleSoft operator has marked the process request for deletion.
ERROR	FAIL	The attempt to launch failed for some reason. In this case the process request never actually started execution.
HOLD	HOLD	The process is ineligible for launching. This may be due to unsatisfied dependencies.
QUEUED	READY	Ready to launch.
INITIATED	INIT	Attempting to launch.
PROCESSING	EXEC	Process is executing.
CANCELLED	CANCEL/ABEND*	Process cancelled.
SUCCESSFUL	SUCC	Successful execution.
NOT SUCCESSFUL	ABEND	Execution failed.

Notes:

- Processes that are cancelled in PeopleSoft will be aborted if they are already executing. The IBM Tivoli Workload Scheduler status mapping must reflect this. A process cancelled that has never been initiated should be CANCEL in Tivoli Workload Scheduler. A process that has been initiated should be ABEND.

2. A process request marked for deletion in the process scheduler can never be eligible to launch. The record is cleared out of the table during a regular maintenance process. For purposes of jnextday processing, such requests should be considered non-existent.

Part 4. R/3 Access Method

Chapter 13. Introduction

IBM Tivoli Workload Scheduler for Applications for R3 gives you the ability to create, schedule, and control R/3 jobs using the job scheduling features of Tivoli Workload Scheduler.

Features

With IBM Tivoli Workload Scheduler for Applications for R3, you can do the following:

- Use Tivoli Workload Scheduler standard job dependencies and controls on R/3 jobs.
- Create R/3 jobs using the Tivoli Workload Scheduler interface.
- Schedule R/3 jobs to run on specified days and times, and in a prescribed order.
- Define inter-dependencies between PeopleSoft jobs and Tivoli Workload Scheduler jobs that run on different applications such as PeopleSoft and Oracle.
- Define inter-dependencies between R/3 jobs and jobs that run on different platforms.
- Define the National language support options.
- Define how to utilize the R/3 Business Warehouse Support functionalities.
- Customize job execution return codes.
- Use R/3 logon groups for load balancing and fault-tolerance.
- Work with R/3 variants and placeholders.
- Exploit BC-XBP 2.0 interface support.

Scheduling Process for the R/3 Extended Agent

Tivoli Workload Scheduler launches jobs in R/3 by way of jobs defined to run on a Tivoli Workload Scheduler extended agent workstation. An R/3 extended agent workstation is defined as a Tivoli Workload Scheduler workstation that is hosted by an FTA or master workstation which uses the `r3batch` access method. The R/3 extended agent workstation uses the access method `r3batch` to pass R/3 job-specific information to predefined R/3 instances. The access method uses information provided in an options file, discussed further in this guide, to connect and launch jobs on an R/3 instance. Numerous extended agent workstations may be defined to use the same host; this is done using multiple options entries or multiple options files. R/3 jobs defined in Tivoli Workload Scheduler must correspond to existing jobs in the R/3 environment. Using the R/3 extended agent name as a key, `r3batch` uses the corresponding options file to determine which instance of R/3 will run the job. It makes a copy of a template job in R/3 and marks it as able to run with a start time of `start immediate`. Finally, it monitors the job through to completion, writing job progress and status information to a job standard list found on the host workstation.

For more information about job management, refer to the *IBM Tivoli Workload Scheduler Reference Guide SH19-4556*.

XBP 2.0 Interface Support

Tivoli Workload Scheduler for Applications for R3 fully supports the XBP 2.0 Interface. This allows for the following advantages:

- Tracking of child jobs.
A Tivoli Workload Scheduler job remains in EXEC status until all its R/3 job tree has completed. The job logs of the R/3 jobs are retrieved and stored in the Tivoli Workload Scheduler `stdlist`, together with the spool lists of all their ABAP steps.
- Interception of R/3 jobs.
Jobs scheduled by R/3 users can be intercepted before execution based on job attributes. An Interception Collector job is scheduled periodically as a Unix or Windows command. In each run, it collects intercepted R/3 jobs and submits them to the current plan based on a template.
- All job attributes are maintained when re-running jobs.
Limitations involving print/archive parameters and job class in conjunction with the use of old copies of re-running jobs are removed.
- External command steps.

See “Using XBP 2.0 Support” on page 95 for details.

Chapter 14. Installation

This section contains information on installing IBM Tivoli Workload Scheduler for Applications for R3.

Software Requirements

To install the IBM Tivoli Workload Scheduler for Applications for R3, you must be running the following:

- IBM Tivoli Workload Scheduler Version 7.0 or later or IBM Tivoli Workload Scheduler for z/OS with end to end.
- R/3 Version 3.1g, 3.1h, 3.1i, 4.0x, 4.5x, 4.6x, 4.7, 6.10, or 6.20.
- For versions earlier than 6.20, the appropriate R/3 Support Package as stated in the following table:

Table 13. R/3 Support Packages.

R/3 Versions	R/3 Notes
R/3 3.1x	393708, 430087, 357924, 517672
R/3 4.0x	393708, 430087, 111696, 357924, 517672
R/3 4.5x	148436, 172245, 326202, 326752, 393708, 399449, 357924, 517672
R/3 4.6x	326752, 352606, 399449, 453135, 435111, 357924, 517672, 610480
R/3 4.7	610480
R/3 6.10	399449, 435111, 453135, 357924, 517672, 610480

Note: R/3 notes 399449 and 430087 resolve print parameter issues. Notes 357924, 517672 and 519137 are Release-independent.

Installation Overview

The following is an overview of the Tivoli Workload Scheduler for Applications for R/3 access method installation steps. Before installing the access method, ensure that Tivoli Workload Scheduler is properly installed on the host workstation.

1. On the system that will be the host of the extended agent for R/3 perform the following steps:
 - a. If the platform where you are installing is Compaq Tru64, follow the instructions described in “Installing on Tier2 Platforms” on page 86. Otherwise, follow the instructions described in Chapter 2, “Installing IBM Tivoli Workload Scheduler for Applications”, on page 5.
2. Install the **r3batch** ABAPs on the R/3 system. To do this:
 - a. Log in to the R/3 instance as a Basis administrator or user with the correct authorizations.
 - b. Import the correction and transport files.

Installing on Tier2 Platforms

1. Log on as root and change to the *TWSHOME* directory.
2. From the installation CD-ROM, extract the software:

```
tar xvf cd_path/TWS4APPS_R3BATCH_OSF.TAR
```

where *cd_path* is the pathname of your CD-ROM drive.
3. Verify that **r3setup** and **r3btar.Z** have been extracted successfully in the current directory.
4. Run the **r3setup** script to decompress the files, perform the initial setup, and create the *r3options* file:

```
/bin/sh r3setup -new|-update
```

where:

-new

Specifies a new installation.

-update

Specifies that you are installing over a previously installed version.

You will be prompted to specify the information listed below. See Chapter 20, "R/3 Access Method Options Files", on page 117 for more details.

- The Tivoli Workload Scheduler user name if different from the default value (default: *maestro*).
- The directory where the extended agent should be installed (type the path to *TWSHOME*).
- The *Launch User* *LJuser* (default: *maestro*). This is the Tivoli Workload Scheduler user name that runs *r3batch*. This must be a valid UNIX account.
- The *Info User* *IFuser* (default: *maestro*). This is the Tivoli Workload Scheduler user name that retrieves job information. *IFuser* must be the same as the *LJuser*.
- The Job definition type *JobDef* (default: *r3batch*). This is the Tivoli Workload Scheduler job definition interface type. It can be:
 - a. *r3batch* (default). It provides a job definition interface that is specific to R/3 jobs.
 - b. *maestrogen* or blank. It provides the standard job definition interface with no R/3 specific entry field.
- The host CPU name for the Tivoli Workload Scheduler extended agent definition.
- The host name of your R/3 application server.
- If you want to specify a separate R/3 gateway host, type *NO*.
- The three-character R/3 system id (for example: *OSS* or *TRN*).
- The two-digit R/3 instance number (for example: *00* or *01*).
- The three-digit R/3 client number (for example: *100* or *900*).
- The R/3 user id for RFC calls (for example: *rfcuser*)
- The password for your R/3 RFC user id.
- The R/3 interface audit level (default: *0*)
- If you wish to define the SAP R/3 login language for extended agent on this R/3 system (default *No*)
- If you wish to define another instance of the extended agent. If you reply *Yes*, you are asked to provide another set of parameters. An additional entry is inserted in the R/3 options file.

5. Open the Tivoli Job Scheduling Console or Composer program and create a workstation definition in Tivoli Workload Scheduler for each instance of R/3.

Encrypting R/3 User Passwords

When you add your entries in your options file, the *r3pass* 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. Run the encryption command as follows:

```
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.

Chapter 15. R/3 Configuration

To communicate and control job execution on R/3 systems using the access method for R/3, you need to perform the following configuration steps on the R/3 environment.

The steps described in this chapter will require an R/3 BASIS Administrator.

Overview

Here is an overview of the customization procedure:

1. Create a new RFC user ID in 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 R/3 server.
4. Import the correction and transport files into R/3 and verify the installation.

Note: The import procedure only adds new ABAP/4 function modules and several new internal tables to the 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 R/3, you must create a user ID in 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** or **Communications**, depending on the R/3 release.
 - A password at least six characters in length. Tivoli Workload Scheduler requires this password to start or monitor R/3 jobs. If this password changes in R/3, you must update the options file used by r3batch with the new password.
 - Assign the appropriate security profiles, depending on your version of R/3.

Creating 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

Do the following:

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_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_PROGRAM	ABAP: program run checks	S_ABAP_ALL
S_RFC	Auth. check for RFC access	S_RFC_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 R/3, it is possible that the authorization S_RFC_ALL is located 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 window and assign the profile to the Tivoli Workload Scheduler R/3 user.
5. Save the user data.

Using Transaction PFCG (Profile Generator)

Do the following:

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

Object	Description
S_ADMI_FCD	System authorizations
S_BTCH_ADM	Background processing: Background administrator
S_BTCH_JOB	Background processing: Operations on background jobs

Object	Description
S_BTCH_NAM	Background processing: Background user name
S_PROGRAM	ABAP: Program run checks
S_RFC	Authorization check for RFC access
S_SPO_ACT	Spool: Actions
S_SPO_DEV	Spool: Device authorizations
S_XMI_LOG	Internal access authorizations for XMI log
S_XMI_PROD	Auth. for external management interfaces (XMI)

8. Fill the values according to the following scheme:

S_ADMI_FCD	System authorizations <ul style="list-style-type: none"> System administration function: Full authorization
S_BTCH_ADM	Background processing: Background administrator <ul style="list-style-type: none"> Background administrator ID: Full authorization
S_BTCH_JOB	Background processing: Operations on background jobs <ul style="list-style-type: none"> Job operations: Full authorization Summary of jobs for a group: Full authorization
S_BTCH_NAM	Background processing: Background user name <ul style="list-style-type: none"> Background user name for authorization check: Full authorization
S_PROGRAM	ABAP: Program run checks <ul style="list-style-type: none"> User action ABAP/4 program: Full authorization Authorization group ABAP/4 program: Full authorization
S_RFC	Authorization check for RFC access <ul style="list-style-type: none"> Activity: Full authorization Name of RFC to be protected: Full authorization Type of RFC object to be protected: Full authorization
S_SPO_ACT	Spool: Actions <ul style="list-style-type: none"> Authorization field for spool actions: Full authorization Value for authorization check: Full authorization
S_SPO_DEV	Spool: Device authorizations <ul style="list-style-type: none"> Spool - Long device names: Full authorization
S_XMI_LOG	Internal access authorizations for XMI log <ul style="list-style-type: none"> Access method for XMI log: Full authorization
S_XMI_PROD	Authorization for external management interfaces (XMI) <ul style="list-style-type: none"> 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 you wrote in **Role Name**.
11. Exit the authorization management window and select the **User** tab.
12. Add the Tivoli Workload Scheduler user to the role.
13. Save the role.

Copy the Correction And Transport Files

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

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

```
copy controlfile /usr/sap/trans/cofiles/
```

```
copy datafile /usr/sap/trans/data/
```

The names of *controlfile* and *datafile* vary from release to release. The files are installed in the *TWSHome*\methods directory and have the following file names and format: K000xxx.TV1 (control file) and R000xxx.TV1 (data file), where *xxx* is a value related to the release.

Import ABAP/4 Modules into R/3

This procedure generates, activates, and commits new ABAP/4 function modules to your R/3 system and several new internal tables. No existing R/3 system objects are modified.

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
```

is the transport request file.

```
sid
```

is your R/3 system ID.

Under the *TWSHome*\methods directory, you can find a file named K000xxx.TV1, so the transport request is tv1K000xxx.

3. Run the **tp tst** command to test the import:

```
tp tst transport sid
```

After you have run this command, examine the log files in the /usr/sap/trans/log directory for error messages. Warnings of severity level 4 are normal.

If you have 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 R/3 database. They automatically become active.

After you have run this command, examine the log files in the /usr/sap/trans/log directory for error messages. Warnings of severity level 4 are normal.

If a problem is encountered, use unconditional mode when executing this step:
`tp import transport sid U126`

5. When the import is complete, check the log files to verify that the import was successful. The log files are in the `/usr/sap/trans/log` directory.

Installing the r3batch ABAP Modules on the R/3 System

Install r3batch ABAP modules on the R/3 system. For details about installing the r3batch ABAP modules see Chapter 15, “R/3 Configuration”, on page 89. This Chapter includes instructions for creating an RFC user and installing the following ABAP modules:

```
J_101_BDC_STATUS
J_101_DATE_TIME
J_101_JOB_ABAP_STEP_MODIFY
J_101_JOB_ADD_ABAP_STEP
J_101_JOB_ADD_EXTPGM_STEP
J_101_JOB_CLOSE
J_101_JOB_COPY
J_101_JOB_DEFINITION_GET
J_101_JOB_DELETE
J_101_JOB_EXTPGM_STEP_MODIFY
J_101_JOB_FIND
J_101_JOB_FINDALL
J_101_JOB_HAS_EXTENDED_VARIANT
J_101_JOB_LOG
J_101_JOB_LOG_READ
J_101_JOB_OPEN
J_101_JOB_START
J_101_JOB_STATUS
J_101_JOB_STATUS_CHECK
J_101_JOB_STOP
J_101_RAISE_EVENT
J_101_REPORT_ALL_SELECTIONS
J_101_REPORT_DYNSEL_GET_DEF
J_101_REPORT_GET_TEXTPOOL
J_101_TABLE_GET_NAMETAB
J_101_VARIANT_COPY
J_101_VARIANT_COPY_4X
J_101_VARIANT_CREATE
J_101_VARIANT_CREATE_NEW
J_101_VARIANT_DELETE
J_101_VARIANT_DELETE_NEW
J_101_VARIANT_EXISTS
J_101_VARIANT_GET_DEFINITION
J_101_VARIANT_GET_DEF_NEW
J_101_VARIANT_GET_HELP_VALUES
J_101_VARIANT_INFO_GET
J_101_VARIANT_MAINTAIN_CNT_TBL
J_101_VARIANT_MAINTAIN_SEL_TBL
J_101_VARIANT_MODIFY
J_101_VARIANT_MODIFY_NEW
```

Troubleshooting

If you are unable to submit R/3 jobs using Tivoli Workload Scheduler after the R/3 configuration, perform the following tests:

- Ensure you can ping the R/3 system from the Tivoli Workload Scheduler system. This will show basic network connectivity.
- Note that using the 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 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 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 R/3 server and *xx* is the R/3 instance.

- Log on to the 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 89) exists.
- Create a new IBM Tivoli Workload Scheduler job to be run under R/3, and add the following advanced options:

```
-debug  
-trace
```

These options are selected using the radio buttons in the Properties - Job Definition Task page in the Job Scheduling Console.

Using the `-debug` option, additional debug information is included in the `stdlist` file (in Tivoli Workload Scheduler).

When you use the `-trace` option, a trace file is created in the Tivoli Workload Scheduler home directory. On UNIX, this trace file is called `dev_rfc`. On Windows, the file is called `rfcxxxxx_xxxxx.trc`. Ensure you delete the trace option from the job after you have performed debug procedures. The trace file can become very large and unmanageable.

Note: Typically these options are for debugging the extended agent and should not be used in standard production.

- If for some reason the R/3 gateway truncates the connection string, replace the IP name with the IP address.
- If you are installing on an AIX system that does not use U.S. English, you must install the U.S. Language Environment on the IBM Tivoli Workload Scheduler workstation and the 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 R/3 will fail.

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.

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

On 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 `TWSHOME/methods`. To do this in a command shell, type:

```
enigma newpass
```

where *newpass* 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 `TWShome/methods` directory. The file can be edited with any text editor.

Ensure you copy the password exactly, preserving upper case, lower case, and punctuation. The encrypted password looks similar to:

```
#TjM-pYm#-z82G-rB
```

If the encrypted password is mistyped, Tivoli Workload Scheduler will not be able to start or monitor R/3 batch jobs.

Migrating from Previous Versions

This version of the Tivoli Workload Scheduler extended agent for R/3 supports all the R/3 versions listed in Table 13 on page 85. To achieve R/3 XBP certification for R/3 versions 4.0 and 4.5, a few changes had to be made to the scheduling agent. The Tivoli Workload Scheduler extended agent for R/3 version 4.x uses the official R/3 RFC interfaces for job scheduling. These are:

- The SXMI_XBP interface for R/3 versions 3.1g to 4.0x
- The BAPI_XBP interface since version 4.5a

To avoid conflicts with other vendors, the Tivoli Workload Scheduler ABAP modules now belong to the Tivoli Workload Scheduler partner namespace `J_1O1_xxx`. After you have completed the imports as described in “Import ABAP/4 Modules into R/3” on page 92, the RFC `J_1O1_xxx` function modules are installed on your system.

If you had a previous installation of Tivoli Workload Scheduler extended agent for R/3 on your system, you can delete the following function modules from your 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_OPEN  
Z_MAE2_JOB_LOG  
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 you should do so if you want to clean up your system.

Using XBP 2.0 Support

The SAP interface for external job scheduling (BC-XBP) has been recently extended in order to remove the limitations reported by customers and partners. It is delivered by Basis Support Packages for all R/3 releases starting from WAS 4.6B.

The following limitations have been removed:

- Job class is no longer limited to C
- All print parameters are available

The following features have been added:

- Parent-child relationship
- Job interception
- Support for jobs with external command steps

Check and apply following SAP notes prior to using the XBP 2.0 interface:

- 602683
- 612564
- 604496

Creating The Authorization Profile for the Tivoli Workload Scheduler User

There are two alternative ways to add the related authorization object to the authorization profile of the Tivoli Workload Scheduler user:

- Using Transaction SU02
- Using the Profile Generator

Using Transaction SU02

Add the following object:

Object	Description	Authorization
S_RZL_ADM	CCMS: System Administration	S_RZL_ALL

Using the Profile Generator

Do the following:

1. Manually add the following object:

Object	Description
S_RZL_ADM	CCMS: System Administration

2. Set the value of the **activity** field to Full authorization.

Print Parameter and Job Class Issues

The extended agent for R/3 uses the official RFC interfaces of R/3 Versions 4.x to 6.20 for job scheduling. When you migrate from previous versions of 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.

The following table shows the print parameters supported by BAPI XBP 1.0.

Print Parameter	3.1x	4.0x	4.5x and later
output device	✓	✓	✓
print immediately	✓	✓	✓
delete after output	✓	✓	✓
number of copies	✓	✓	✓
archiving mode	✓	✓	✓
authorization	✓	✓	✓
selection cover page	x	✓	✓

Print Parameter	3.1x	4.0x	4.5x and later
sap cover page	✓	✓	✓
spool retention period	✓	✓	✓
recipient	x	x	✓
lines	x	x	✓
columns	x	x	✓

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

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

Chapter 16. Defining R/3 Extended Agent Jobs

To control job execution on the R/3 workstation from Tivoli Workload Scheduler you must define the following:

- Jobs in R/3 that you want to run under Tivoli Workload Scheduler control. You can define these jobs using standard R/3 tools, or the SAP-specific windows of the Job Scheduling Console.
- Jobs in Tivoli Workload Scheduler that correspond to the jobs in R/3. The Tivoli Workload Scheduler job definitions are used in scheduling and defining dependencies, but the R/3 jobs are actually run.

Defining a Job in R/3

To create R/3 jobs that are controlled by Tivoli Workload Scheduler, use either of the following:

- The standard R/3 tools for defining jobs
- The SAP-specific windows of the Job Scheduling Console.

Refer to the R/3 documentation for information on creating jobs with R/3 tools. When creating jobs in R/3, do not assign a start time and do not specify any dependencies. These are specified and controlled in Tivoli Workload Scheduler. The job is identified in Tivoli Workload Scheduler by its R/3 job name and job ID. The same job can be scheduled to run repeatedly by Tivoli Workload Scheduler without having to redefine it each time in R/3.

Note: For R/3 jobs controlled by Tivoli Workload Scheduler, the job log output in R/3 is found under the user name defined in the *r3user* option of the options file.

Defining a Job in Tivoli Workload Scheduler

To define the Tivoli Workload Scheduler job, you can use one of the following:

- The Tivoli Job Scheduling Console.
- The Tivoli Workload Scheduler command line.
- The Tivoli Workload Scheduler for z/OS ISPF dialogs, if you plan to schedule the job in an end-to-end environment.

See Chapter 5, “Defining Extended Agent Jobs”, on page 35 for reference.

The procedures described in Chapter 5, “Defining Extended Agent Jobs”, on page 35 are common to all the Tivoli Workload Scheduler extended agents. The extended agent for R/3 provides an additional one through the Job Scheduling Console.

In the Job Scheduling Console, when you select the task type for a new job definition, you can choose between two types:

Extended Agent Task

It is common to all extended agents. If you choose this option:

- The associated R/3 job must be already defined in the R/3 database and you must know its name.

- You can use the **Command** field in one of the job definition windows to enter a number of parameters for running the R/3 job.
- The extended agent workstations you defined for R/3 (specifying `r3batch` as the access method) are listed as generic extended agent workstations.

SAP It is specific to R/3 extended agent jobs. If you choose this option:

- You can define or update the R/3 job, as well as the SAP variants or parameters for running the job, from the Job Scheduling Console windows as if you were operating on R/3.
- The R/3 extended agent workstations are considered R/3 workstations. They do not show in the generic extended agent workstations lists that are displayed when you click the ellipsis next to the **Workstation** field. You must already know the name to write it in a field.

To have **SAP** displayed as an optional task in the Job Scheduling Console, you must set the `jobdef` option to **r3batch** in the `r3batch.opts` options file (located in the `methods` subdirectory of `TWSHome`) of the Tivoli Workload Scheduler master.

To learn how to define an R/3 extended agent job using the Extended Agent Task windows, see Chapter 5, “Defining Extended Agent Jobs”, on page 35. To learn how to define an R/3 extended agent job using the SAP task windows, see the next section.

Using the SAP-Specific Windows of the Job Scheduling Console

To create a job definition in the Tivoli Workload Scheduler database for an R/3 job, perform the following steps:

1. In the Actions list of the Job Scheduling Console main window, select **New Job Definition**.
2. Select a Tivoli Workload Scheduler engine and from the ensuing list click **SAP**. The Properties - Job Definition window is displayed.

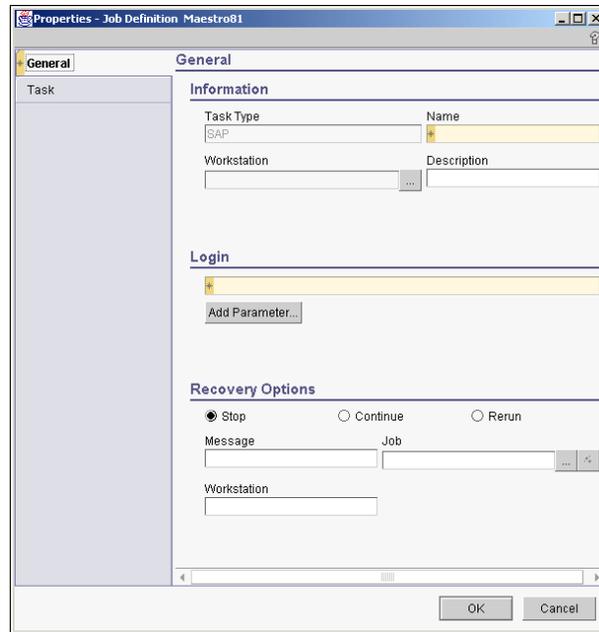


Figure 26. Properties - Job Definition

3. In the General page you define the Tivoli Workload Scheduler job that will be driving the R/3 job. Complete the following fields:
 - a. In the **Name** field, enter a Tivoli Workload Scheduler name for the R/3 job. The name can contain eight alphanumeric characters, or 40 alphanumeric characters if you are using the expanded database option. The job name must start with a letter.
 - b. In the **Workstation** field, click the ellipsis (...) button to open a Find window and select an available workstation.
 - c. In the **Description** field, enter a description for the job. This field is an optional text description of the job and can be up to 64 alphanumeric characters.
 - d. In the **Login** field, enter the Tivoli Workload Scheduler log on ID that is used to run the job.
 - e. Click **Add Parameter** to add any predefined parameters to the log on ID.
 - f. In the **Recovery Options** fields, specify any recovery options for the R/3 job. Refer to the *IBM Tivoli Workload Scheduler Reference Guide* for information about recovery options.
4. Select **Task**. The Task page is displayed.

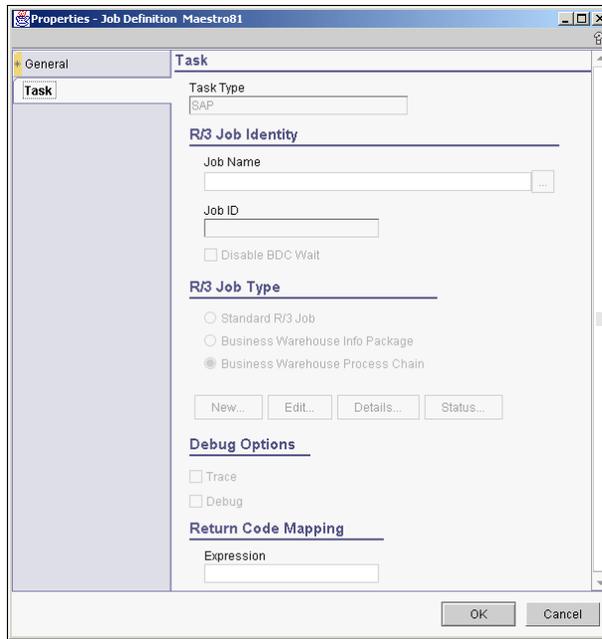


Figure 27. Properties - Job Definition Task

In this page, you enter the specifications of the R/3 job that will be driven by the Tivoli Workload Scheduler job you defined in the previous page. You can use this page also to create standard R/3 jobs (but no InfoPackages or Process Chains) in the SAP system .

5. To specify an existing R/3 job:
 - a. In the Task page, specify the job type and click the ellipsis (...) next to the **Job Name** field. The SAP Pick List window is displayed.
 - b. Use the pick list to find and select an R/3 job and click **OK**. The job information is displayed in the following items back in the Task page:
 - Job Name field
 - Job ID field
 - R/3 Job Type button

If the job type is Standard R/3 Job, you can edit it and see its details and status.

If the job type is Business Warehouse Info Package or Process Chain, you can only see its details.

6. To create a standard R/3 job:
 - a. In the Task page, select **Standard R/3 Job** and click **New**. The R/3 Job page of the SAP Job Definition window is displayed.

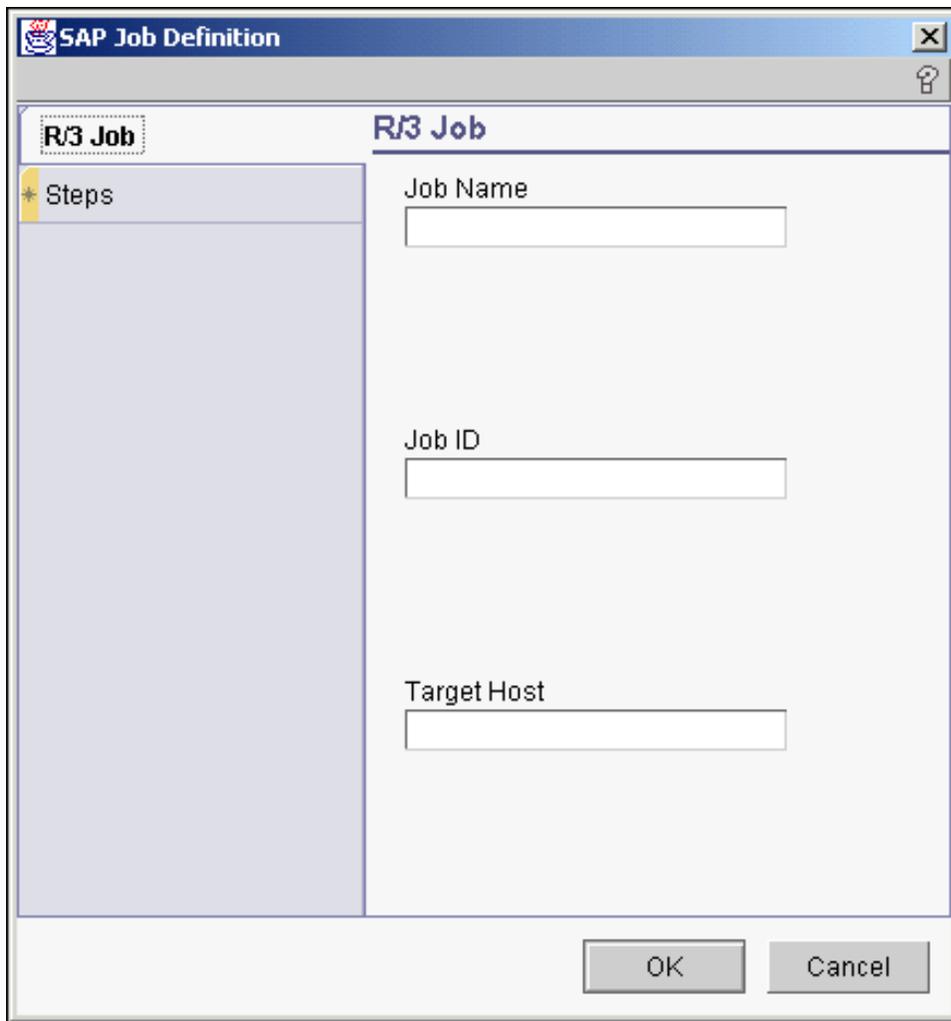


Figure 28. The R/3 Job page of the SAP Job Definition window.

- b. In the R/3 Job page, enter information about the job you are creating:
 - 1) Type a name for the R/3 job in the **Job Name** field.
 - 2) Leave the **Job ID** field blank. An ID will be assigned by the R/3 system when you save the job definition.
 - 3) Type the name of the R/3 workstation where this job will run in the **Target Host** field.
 - 4) Click **Steps** to open the following page.

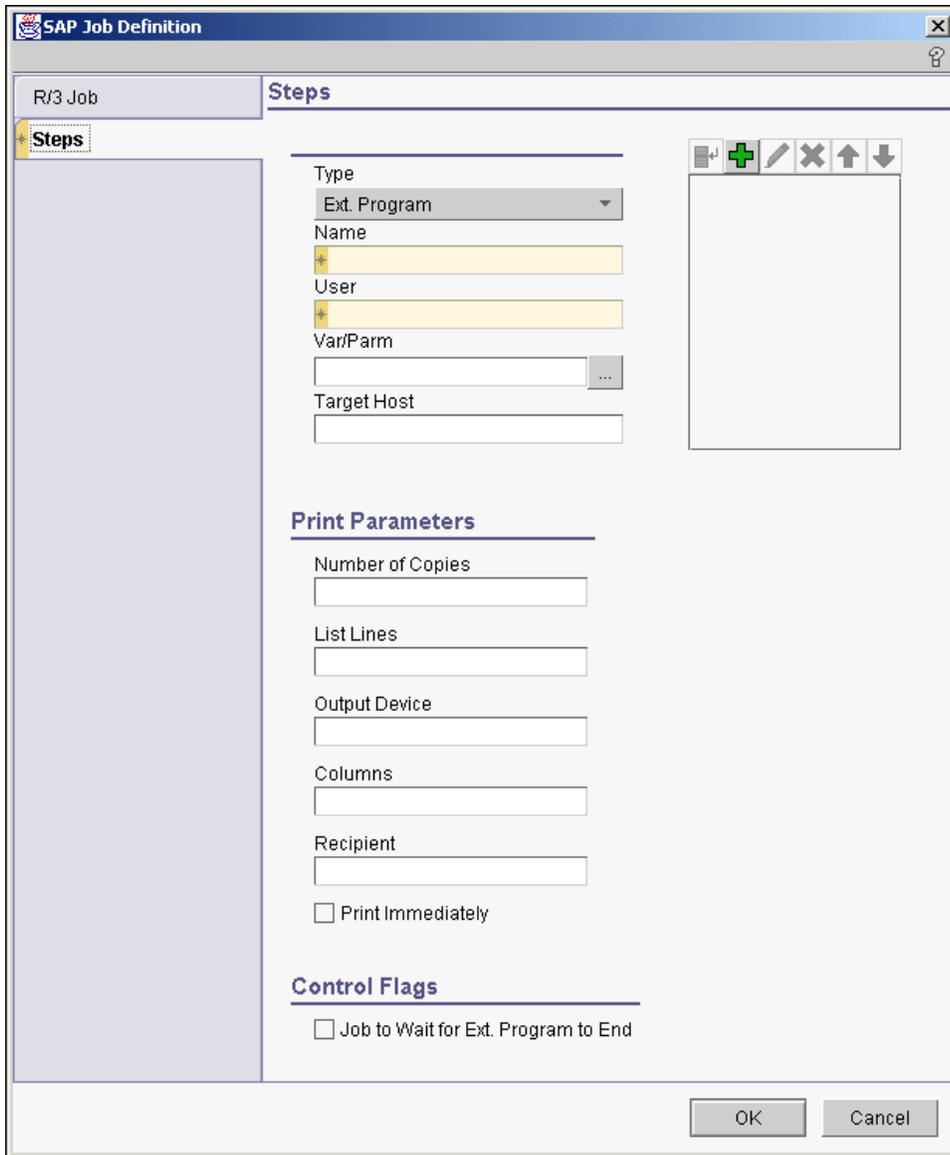


Figure 29. The Steps page of the SAP Job Definition window.

- c. In the Steps page you specify one or more ABAPs or external programs for the R/3 job. Repeat the following steps for as many steps as you have to define:
 - 1) Click the **Add row** button to add a row to the steps table.
 - 2) Double-click in the **Type** field and select whether this is to be an ABAP or external program.
 - 3) In the **Name** field, enter an ABAP name or a fully qualified path and file name for an external program.
 - 4) In the **User** field, type the name of the R/3 user who runs this step.
 - 5) In the **Var/Parm** field, type a variant name or a parameter, if necessary. Variants are used with ABAPs, and parameters are used with external programs, but both are optional. Not all ABAPs require variants and not all external programs require parameters. An ellipsis (...) button is available next to the field to display a list of the variants associated with the ABAP. You can use this list to select a variant as well as for creating or changing variant definitions. See “Defining or Updating R/3 Variants from the Job Scheduling Console” on page 105 for details.

- 6) In the **Target Host** field, enter the R/3 workstation where this step runs. Target hosts are only required for external programs.
 - 7) Enter print parameters and control flags specifications according to R/3 requirements.
- d. When you have finished creating the ABAP or external program steps, click **OK**. The R/3 job definition is saved in the R/3 database and the window is closed. If the R/3 job was successfully saved in the R/3 database, the Properties - Job Definition Task page should display an R/3 job ID.
7. Click **OK** in the Properties - Job Definition window to save the job definition in the Tivoli Workload Scheduler database.

Defining or Updating R/3 Variants from the Job Scheduling Console

The action of clicking the ellipsis alongside the **Var/Parm** field in the SAP Job Definition window displays a Variant List window. This window lists all the variants associated with the ABAP featured in the **Name** field in SAP Job Definition and resembles the one pictured in the following figure.

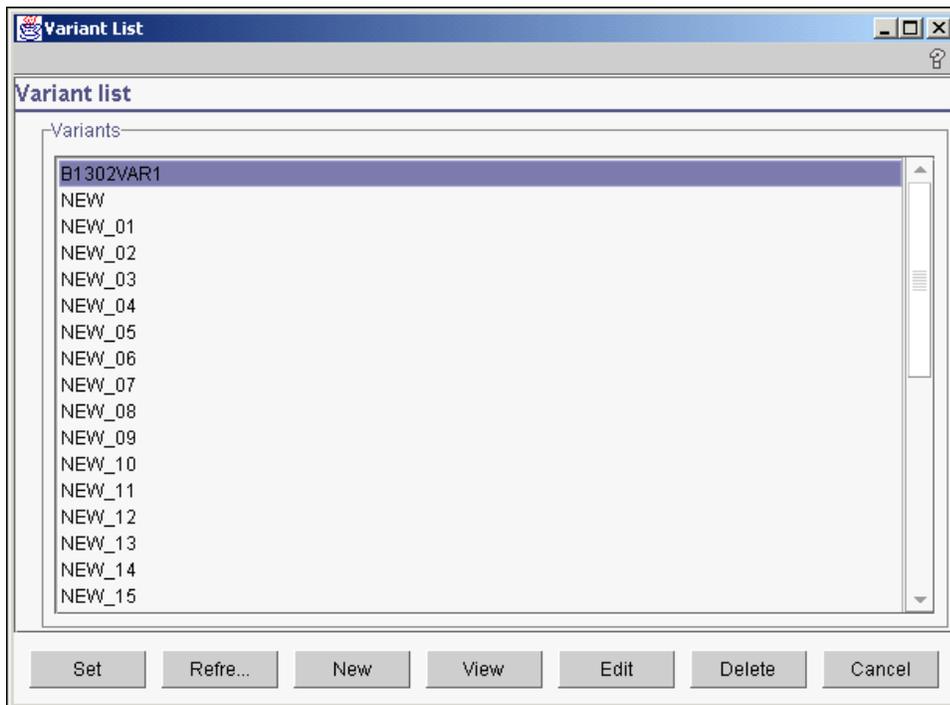


Figure 30. The Variant List window.

From this window, in addition to other actions, you can create or update variants. To do this, click **New** or **Edit**. A Variant Frame window similar to this is displayed (if you are editing, the fields and selections will not be empty):

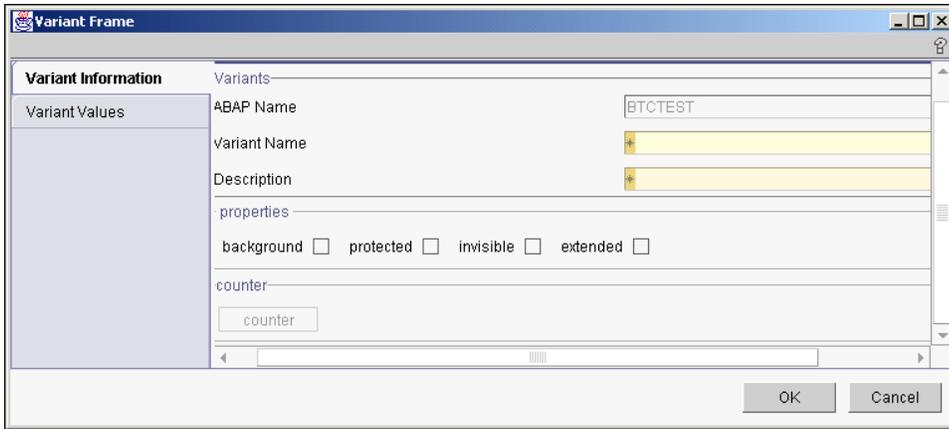


Figure 31. The Variant Information page of the Variant List window.

In the Variant Information page do the following:

1. Enter/modify the variant name and description.
2. 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 in the SAP GUI.

Extended

Allows for the use of placeholders and counters as variant values. If you check this box, the **counter** button becomes available.

For extended variants, you can use placeholders and counters that eliminate the error-prone task of adjusting values and thus dramatically minimize the effort for variant maintenance. Placeholders and counters are pre-processed by Tivoli Workload Scheduler and the values are automatically adjusted when the job is launched. Supported placeholders and counters are:

Table 14. 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
\$Nx	Counters	10 counters: \$N0 - \$N9 (\$N = \$N0)

Table 14. Placeholders and counters for extended variants. (continued)

Symbol	Meaning	Syntax
\$(date expression)	Date expression	Like datecalc command. Enclosed within \$(and).
\$(arithmetic expression)	Arithmetic expression	Arithmetic expressions allowing for +, -, *, and / operations between integers and counters.

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 contained in the equivalent SAP window.

Task String Parameters for R/3 Jobs

This section describes the task string parameters that rule the execution of R/3 jobs. You must specify them in the following places when you define their associated Tivoli Workload Scheduler jobs:

- In the **Command** field of the Task page of the Properties - Job Definition Window, if you use the Job Scheduling Console and selected **Extended Agent Task** (and not **SAP**) for a new 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 string syntax is the following:

```
-job job_name ( -user user_name | -i (job_id|ipack_|pchain_) )
  [-client def_client] [-exec_client exec_client] [-debug] [-trace]
```

The parameters are described in the following table.

Table 15. Task String Parameters for R/3 Jobs.

Parameter	Description
-job job_name	The name of the task that is to be run. It is either an R/3 job name, InfoPackage technical field name, or ProcessChain name. This parameter is mandatory.
-user user_name	The R/3 user who owns the target job. This parameter can be used only when the target R/3 system has only one job with the specified name for the specified user. It has no effect with InfoPackage or ProcessChain jobs, nor if a job id has been specified in the job definition.

Table 15. Task String Parameters for R/3 Jobs. (continued)

Parameter	Description
<code>-i job_id</code>	<p>One of the following:</p> <ul style="list-style-type: none"> • The unique R/3 job id, if the target job is a standard one. • <code>ipack_</code>, if the target job is an InfoPackage. • <code>pchain_</code>, if the target job is a ProcessChain. <p>The <code>-user</code>, <code>-client</code>, and <code>-exec_client</code> parameters will have no effect, if you specify a job id. For InfoPackage and ProcessChain jobs, this parameter is mandatory.</p>
<code>-client def_client</code>	<p>The number identifying the R/3 client where the job definition is to be found, regardless of the client number defined by the <code>r3client</code> key in the options file. This parameter has no effect with InfoPackage or ProcessChain jobs, nor if a job id has been specified in the job definition.</p>
<code>-exec_client exec_client</code>	<p>The number identifying the R/3 client where the job is to be run, regardless of the client number defined by the <code>r3client</code> key in the options file. This parameter has no effect with InfoPackage or ProcessChain jobs, nor if a job id has been specified in the job definition.</p>
<code>-debug</code>	<p>Turns on the most verbose r3batch trace.</p>
<code>-trace</code>	<p>Turns on the SAP RFC trace.</p>

Notes:

1. You can specify both `-i` and `-user` in the same job definition, but the user name will be ignored.
2. When you specify the job id, `-client` and `-exec_client` are ignored since the id is unique for the entire R/3 system.

The following is an example for an R/3 job named BVTTEST with id 03102401 and user myuser:

```
-job BVTTEST -i 03102401 -debug
```

The following is an example for an InfoPackage job whose technical field name is ZPAK_3LZ3JRF29AJDQM65ZJB50MY:

```
-job ZPAK_3LZ3JRF29AJDQM65ZJB50MY -i ipak_
```

Tivoli Workload Scheduler and R/3 Job States

When an R/3 job is launched by Tivoli Workload Scheduler, you can monitor its progress. The status transitions in Tivoli Workload Scheduler (internal status) and the corresponding R/3 status are listed in the following table.

Tivoli Workload Scheduler Job State	R/3 Job State
INTRO	N/A
WAIT	Ready
EXEC	Active
SUCC	Finished
ABEND	Canceled

The INTRO state indicates that Tivoli Workload Scheduler is in the process of introducing the job, but in R/3, 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 may last a few minutes if the R/3 system is particularly busy.

Although a job may be finished in R/3, Tivoli Workload Scheduler will keep it in the EXEC state if its BDC sessions are not complete and you have not selected the **Disable BDC Wait** option. To use this option, see Chapter 21, "BDC Wait", on page 125.

Chapter 17. Re-running Jobs

Before you re-run an R/3 job with Tivoli Workload Scheduler, you have the option of providing a step name for the job. When you do so, this affects r3batch in the following way: if the step name has the “xnnnnnn” or “nnnnnnn” format (where “x” is a non-digit character and “n” is a digit character), then it is used as the starting step number for the re-running R/3 job.

In all other cases, that value has no effect on the re-running job. The job is re-run starting from its first step.

Only when r3batch re-runs a job from its first step, either because the user has specified it as the starting step or no starting step has been specified at all, it uses the *new copy* feature, if applicable. If the starting step is greater than one, then r3batch uses the *old copy* to re-run the job.

The section below explains the main difference between the new and the old copy of a re-running job.

Old Copy and New Copy of a Re-running Job

When the access method for R/3 needs to launch a job, it makes a copy of a template job and runs it.

The new copy feature is available for versions 3.1i and later of R/3. It copies a job altogether, preserving steps, job class and all print and archive parameters. It is performed using a new R/3 function module that is part of the R/3 Support Package as stated in the R/3 notes 399449 and 430087.

On the contrary, the old copy feature, based on standard R/3 function modules, creates a new R/3 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 support:

- 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 section “Print Parameter and Job Class Issues” on page 96 to learn how to resolve the issue of lost job class and print and archive parameters.

Chapter 18. Business Information Warehouse

Business Information Warehouse Support - InfoPackages

Business Information Warehouse (BIW) is a data warehouse solution tailored to R/3. It allows business reporting and decision support.

User Authorizations

Tivoli Workload Scheduler for Applications for R3 is able to manage R/3 BIW Info Packages and Process Chains. In order to use the R/3 BIW functionality, you need to define an R/3 Tivoli Workload Scheduler user within R/3 that needs the following additional authorizations:

The authorizations in the table below are required only if you want to utilize the Business Warehouse functionalities of the extended agent.

Object	Text	Authorization
S_DEVELOP	ABAP Workbench	E_ABAP_ALL

Note: In addition to the Z_MAESTRO profile, the profile S_RS_ALL needs to be added to the Tivoli Workload Scheduler user.

Business Warehouse Components

R/3 supports two main BIW 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 actual SAP job whose aim is to load data. Like any other R/3 job, it contains job specific parameters like start time, dependencies, and more.

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 Extended Agent Jobs for BIW

InfoPackages and Process Chains can be created only from the R/3 environment. However, feature level 1.3 of the Job Scheduling Console supports pick lists of InfoPackages and Process Chains, so that you can now define Tivoli Workload Scheduler extended agent jobs also for these objects.

See “Using the SAP-Specific Windows of the Job Scheduling Console” on page 100 to learn how to define extended agent jobs through the SAP windows of the Job Scheduling Console.

See Chapter 5, “Defining Extended Agent Jobs”, on page 35 to learn how to define extended agent jobs through the Extended Agent Task windows of the Job Scheduling Console or through Composer.

See “Task String Parameters for R/3 Jobs” on page 107 to know the task string parameters for specifying InfoPackages and Process Chains when defining an extended agent job through the Extended Agent Task windows of the Job Scheduling Console or through Composer.

R/3 Schedule Options of an InfoPackage

In order to be able to schedule InfoPackages through IBM Tivoli Workload Scheduler, the scheduling options of the InfoPackage need to be adjusted like follows:

- Start type has to be set to “Start later in bckgrnd proc.”
- Start time must be “immediate”

Chapter 19. Connecting to the R/3 System

Tivoli Workload Scheduler for Applications for R3 uses the SAP RFC (remote connection call) library to connect to the R/3 system. The connection address for an R/3 system is denoted as a connection string.

Connections to a Specific Application Server

In its basic form, a connection string consists of the hostname (or IP name) of an R/3 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 via TCP/IP. Usually, modern companies use more complex network topologies, with a number of small sub-networks, which cannot communicate directly via TCP/IP. In order to support this type of networks, the SAP RFC library supports so called SAP routers, which are placed at the boundaries of the sub-networks and act as proxies. For this type of networks, the connection string is a composite of basic connection strings for each SAP router, followed by the basic connection string for the target R/3 system; for example:

```
/H/litespeed/H/amsaix33/H/hemlock.tivlab.raleigh.ibm.com
```

Moreover, it is possible to secure the SAP routers with passwords, to prevent unauthorized access. In this case, the basic connection string for this 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 hostnames, without the domain name whenever it is possible. Alternatively, you can use the IP addresses, but this is not recommended, since it is very hard to maintain.

Tivoli Workload Scheduler for Applications for R3 supports both types of connection strings, basic and composite, where:

r3host Is the connection string.

r3instance

Is the R/3 instance number.

r3sid Is the R/3 system ID.

For example:

```
r3host=/H/litespeed/H/amsaix33/H/hemlock.tivlab.raleigh.ibm.com  
r3instance=00  
r3sid=TV1
```

Logon Groups

In large R/3 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.

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 R/3 system ID.

For example:

```
r3host=pwdf0647.wdf.sap-ag.de  
r3group=PUBLIC  
r3sid=QB6
```

Chapter 20. R/3 Access Method Options Files

The options files for the R/3 access method are the following:

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.

CPUNAME_r3batch.opts

A configuration file that is specific to each IBM Tivoli Workload Scheduler extended agent workstation - here generically named “CPUNAME” - using 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.

For instance, if you want to define two extended agents named wkst1 and wkst2 that will access two R/3 Systems - SAP1 and SAP2 - with the r3batch access method, then you must define the following three configuration files:

- The global r3batch.opts
- The local file wkst1_r3batch.opts
- The local file wkst2_r3batch.opts

Note: If r3batch finds the local configuration file for an extended agent, then it ignores the information contained in r3options.

Configuration Options

This section describes the options you want to configure in r3batch.opts and in CPUNAME_r3batch.opts.

Global Options

The following table lists the options that you must place in the global configuration file r3batch.opts. These options have effect only when running the Tivoli Workload Scheduler extended agent in a Unix environment.

Note: These options take effect only if they are placed in r3batch.opts.

Table 16. R3batch Global Configuration Options.

Option	Description
job_sem_proj	The project ID for the job semaphore.
var_sem_proj	The project ID for the variant semaphore.
icp_sem_proj	The project ID for the job interception semaphore.

The job_sem_project and var_sem_project options are two advanced settings that enable you to customize the synchronization among jobs. The R/3 system synchronizes access to the database using a lock management system. When Tivoli Workload Scheduler for Applications for R3 is working in the R/3 system, the corresponding ABAP function modules automatically lock and unlock different objects in the database. When attempting to lock an object, the lock management

system allows to wait only a limited period of time before the lock attempt is rejected with a failure. When this happens, the current function module is aborted as well. The maximum period of time in seconds for each lock attempt is specified in the `enque/delay_max` system parameter.

The return code of the function module is too general to distinguish the overrun of the maximum lock wait time from other database errors. So, the access method for R/3 is not able to proceed and reports an error. If this happens during the execution of an R/3 job, the job is put into abort state.

This situation gets worse when Tivoli Workload Scheduler starts a number of jobs simultaneously. Depending on the maximum lock wait time, a number of them will be aborted due to expiration of the lock attempts.

To resolve this problem, Tivoli Workload Scheduler for Applications for R3 uses semaphores on the OS level to synchronize critical ABAP function module calls. This avoids the described problem effectively. It uses one semaphore for job-related tasks, and another one for tasks related to variant maintenance.

In order to rendezvous on the same IPC structure in a UNIX environment, the communication partners have to agree on an identifier for this structure. There are several ways to choose this identifier. Tivoli Workload Scheduler for Applications for R3 uses two parameters: a pathname and a project ID (which is a character value). The pathname parameter is taken from the absolute path to the options file. The project ID is taken from the options described in Table 16 on page 117. If these options are omitted, Tivoli Workload Scheduler for Applications for R3 uses default values, which will work well on the majority of installations.

Local Options

The following table lists the options that you should place in the local configuration files.

Table 17. R3batch Local Configuration Options.

Option	Description
r3client	The R/3 client number. This option is required.
r3gateway	The host name of the R/3 gateway.
r3group	The name of the R/3 logon group.
r3gwservice	The service number of the R/3 gateway.
r3host	The host name of the R/3 message server when using logon groups, or the host name of the application server in all other cases. This option is required.
r3instance	The R/3 instance number. This option is required.
r3password	The password for r3user . The value is case sensitive. This option is required.
r3sid	The R/3 system ID. This option is required.
r3user	The name of the R/3 user with which the access method connects to the R3 system. This option is required.

Table 17. R3batch Local Configuration Options. (continued)

Option	Description
r3version	Forces the access method to work only with R/3 systems having the specified version. Use this option carefully.
xbpversion	Forces the access method to work with R/3 systems that support the specified XBP version. Use this option carefully.

General Options

The following table lists additional options that you may place in either configuration file.

Table 18. R3batch General Configuration Options.

Option	Description	Default
r3auditlevel	The audit level for the XBP. A number from 0 to 3.	3
short_interval	The short interval, in seconds, for status checks. A value greater than 10 seconds.	10
long_interval	The long interval, in seconds, for status checks. A value smaller than 3600 seconds.	3600
twsxa_cp	The encoding that r3batch is to use to open RFC connections with R/3 systems. Can be one of these values: <ul style="list-style-type: none"> • 1100 • 1103 • 8000 • 8300 • 8400 	1100
twsxa_lang	The language used to login to R/3 systems. It can be: <ul style="list-style-type: none"> • DE for German • EN for English • JA for Japanese 	EN
twsmeth_cp	The encoding that r3batch is to use to write its output. It can be any of the existing TIS codepages.	The local codepage
twsmeth_lang	The language used to report messages.	The local language
utf8cmdline	A value, 0 or 1, that specifies whether r3batch will expect extended parameters encoded in UTF-8. A value of 1 means that it will.	1

Table 18. R3batch General Configuration Options. (continued)

Option	Description	Default
retry	The retry count for R/3 function module calls. A number greater than 0.	5
nojobdefs	Permit/prohibit definition of new R/3 jobs through Tivoli Workload Scheduler GUI.	0
oldcopy	Forces the access method to use the old way of copying jobs. Specify a non-zero value to enable.	0
old_picklist_mode	This option applies only to R/3 versions 4.5 and later. If this option is disabled (value 0), the PickList of jobs is retrieved through the BAPI_XBP_JOB_SELECT function module. If it is enabled (non-zero value), the PickList of jobs is retrieved through the BAPI_XBP_JOB_COUNT function module. This option should be disabled as function module BAPI_XBP_JOB_COUNT executes with problems.	0
rfc_open_retry	The retry count for opening an RFC connection to the R/3 system. The value is a number greater than 0 to limit the number of retries, -1 to have an unlimited number of retries.	5
rfc_open_delay	The maximum number of seconds to wait between two consecutive calls before opening an RFC connection.	1800
log_r3syslog	If enabled, causes the access method to write the latest entries from the R/3 syslog to its trace file when an RFC returns with a general error. Specify ON to enable, OFF to disable.	OFF
rcmap	Enables the RC mapping capabilities of Tivoli Workload Scheduler for Applications for R3. Specify ON to enable, OFF to disable.	ON
max_name_counter	The maximum value of the variant name counter. If the name counter reaches the specified value, it starts over by 0.	40

Table 18. R3batch General Configuration Options. (continued)

Option	Description	Default
name_counter_policy	Name counter policy: step The name counter is increased at every step. job The name counter is increased once for the job.	job
max_n0_counter	Max value of the N0 counter. If the N0 counter reaches the specified value, it starts over from 0.	2 ¹⁵ - 1
n0_counter_policy	N0 counter policy: step The N0 counter is increased at every step. job The N0 counter is increased once for the job.	job

Configuration Options Usage

The format of the `CPUname_r3batch.opts` configuration file is the following:

```
option1=value1
option2=value2
option3=value3
...
```

with no blanks before the option and after the value, nor between them and the equal character.

In `r3batch.opts` you can put all the common information, such as the `LJuser`, `IFuser`, `JobDef` and `LogFileName` options, while you can put in a local configuration file (for example, `wkst1_r3batch.opts`) tailored data for the target R/3 system of the extended agent (for example, `SAP1`).

You can put a local option in the global configuration file if you need to give the same option to all the `r3batch` instances. If R/3 user name is the same in all your R/3 systems, you can place the `r3user` option in the global file without duplicating that information in all the local configuration files.

Note that these configuration files are not created during the installation process.

A global option, like `job_sem_proj`, has effect only in the global configuration file, if you put this option in a local file it will not have any effect.

`r3batch` reads the global configuration file first, then the local file. So every option (except the global options) contained in the local configuration file will override the global's files. For example, if both the global and the local configuration files contain the `r3user` option, `r3batch` will use the local file.

Chapter 21. BDC Wait

With the Batch Data Collector (BDC) wait option, you can specify that an R/3 job launched by IBM Tivoli Workload Scheduler will not be considered complete until all of its BDC sessions have completed. This prevents other IBM 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 completed.

To make use of the option, an R/3 job must write informational messages in its joblog. This can be done by adding a message statement to the SAP-provided report BDCREXX as described below, using the BDC session name (up to 12 characters), or the BDC queue ID (up to 20 characters). Multiple messages can be written to allow monitoring of multiple BDC sessions.

Alternatively, if you do not want to make R/3 modifications, you can also copy BDCREXX to another name, for example ZBDCREXX, make the changes in the copy and include it into all of your programs that generate BDC sessions. You have to do this manually for each job.

If your BDC sessions are not created using transaction SM35, but directly by a SAP component, they may not use report BDCREXX. In this case, you can make modifications to the SAP-provided function module BDC_OPEN_GROUP as described below. Note that you should modify either the BDCREXX report (or its copy ZBDCREXX) or the BDC_OPEN_GROUP function module. You must not modify both objects.

Note: The completion status of a IBM Tivoli Workload Scheduler-launched R/3 job, whether SUCC or ABEND, is based entirely on the execution status of the batch job. It is not related to the success or failure of any BDC sessions, which are considered finished regardless of any errors found during BDC processing. If you want to react on the errors of a BDC session, you can make use of the return code mapping functionality of Tivoli Workload Scheduler as described in Chapter 28, "The Return Code Mapping Library", on page 173.

Using BDC Session Names

In order to use BDC session names, modify the form open_group in the R/3 object BDCREXX as follows:

- **For SAP releases prior to 4.6c:**

```
form open_group.
if session = 'X'.
  skip.
  write: /(20) 'Create group'(i01), group.
  skip.
* open batchinput group
  call function 'BDC_OPEN_GROUP'
  exporting client = sy-mandt
           group = group
           user = user
           keep = keep
           holddate = holddate.
```

```

        message i001(92) with group.
        write: /(30) 'BDC_OPEN_GROUP'(i02), (12) 'returncode:'(i05), sy_subrc.
    endif.
endform.

```

- **For SAP releases 4.6c and later:**

```

form open_group.
    skip.
    write: /(20) 'Create group'(i01), group.
    skip.
* open batchinput group
    call function 'BDC_OPEN_GROUP'
        exporting client = sy-mandt
                group = group
                user = user
                keep = keep
                holddate = holddate.

        message i001(92) with group.
        write: /(30) 'BDC_OPEN_GROUP'(i02), (12) 'returncode:'(i05), sy_subrc.
endform.

```

When BDC sessions are specified by name, Tivoli Workload Scheduler will not wait for the named BDC session to complete if there are other completing BDC sessions of the same name.

Example

An R/3 job launched by Tivoli Workload Scheduler starts three BDC sessions with the name ORD2 and issues the following BDCWAIT message:

```
BDCWAIT:ORD2
```

After the first ORD2 session is complete, the status of the R/3 job is shown as SUCC in Tivoli Workload Scheduler. **r3batch** does not know the difference between non-unique BDCs (that is, named BDCs).

Another problem of using BDC sessions with names is probably different system times in a distributed R/3 environment. **r3batch** scans for BDC sessions with the specified names which are generated during the runtime of the creator job. If the creator job does nothing other than start BDC sessions, the runtime may be very short. If no time synchronization is installed, the system time of different workstations usually differs in a distributed R/3 environment. This time difference may be larger than the job runtime. It is therefore possible that the wrong time interval is scanned and the BDC sessions are never detected.

Therefore it is strongly recommended to use the BDC queue ID method described in the next sections.

Using BDC Queue IDs with Report BDCRECXX

To resolve the problems caused when BDC sessions are specified by name, they can also be specified by queue id, the 20-digit BDC session identifier. Because each R/3-assigned queue id is unique, one BDCWAIT message must be issued for each separate queue id to be monitored.

To determine the queue id, modify the open_group form in the R/3 object BDCRECXX as follows:

- **For SAP releases prior to 4.6c:**

```

form open_group.
if session = 'X'.
  skip.
  write: /(20) 'Create group'(i01), group.
  skip.
* open batchinput group
  call function 'BDC_OPEN_GROUP'
  exporting client = sy-mandt
           group = group
           user = user
           keep = keep
           holddate = holddate
  importing qid = myqid.

  message i001(92) with myqid.
  write: /(30) 'BDC_OPEN_GROUP'(i02), (12) 'returncode:'(i05), sy_subrc.
endif.
endform.

```

- **For SAP releases 4.6c and later:**

```

form open_group.
  skip.
  write: /(20) 'Create group'(i01), group.
  skip.
* open batchinput group
  call function 'BDC_OPEN_GROUP'
  exporting client = sy-mandt
           group = group
           user = user
           keep = keep
           holddate = holddate
  importing qid = myqid.

  message i001(92) with myqid.
  write: /(30) 'BDC_OPEN_GROUP'(i02), (12) 'returncode:'(i05), sy_subrc.
endform.

```

Do not forget to define the *myqid* variable in BDCRECX. With this approach, you obtain a global change in your R/3 system.

Using BDC queue IDs with FM BDC_OPEN_GROUP

BDC sessions not created by the transaction SM35, but by a SAP component, may not use report BDCREXX. In this case, the best solution to track these BDC sessions is to modify 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 i001(92) with qid.
ENDIF.

ENDIF.
*
PERFORM FEHLER_BEHANDLUNG USING SY-SUBRC.
*
*
ENDFUNCTION.
```

Note that the actual parameters of the call of the C function (CALL 'BDC_OPEN_GROUP' ID ...) may vary depending on the SAP release number.

With this approach, you obtain a global change in your R/3 system.

Chapter 22. Job Interception and Parent-child Features

This chapter describes how the Job Interception and Parent-child features of BC-XBP 2.0 are supported by Tivoli Workload Scheduler for Applications.

Job Interception

Job interception is a feature of the BC-XBP 2.0 interface. It enables Tivoli Workload Scheduler to have a very sophisticated control over the jobs launched by R/3 users from the SAP graphical interface.

The job interception mechanism becomes active when the R/3 job scheduler is about to start an R/3 job (that is, the moment when the start conditions of an R/3 job are fulfilled). It checks the job parameters (job name, creator, client) against the entries in the R/3 table TBCICPT1, and when the job parameters match the criteria, the R/3 job is set back to the scheduled status and is marked with a special flag, denoting that the job has been intercepted.

If Tivoli Workload Scheduler has been set up to handle job interception, it periodically runs its own job to retrieve the list of intercepted jobs and reschedules them. This job can be referred to as the interception collector job.

Implement the following steps to set up Tivoli Workload Scheduler to handle job interception in an R/3 environment:

1. Install the BC-XBP 2.0 interface.
2. Activate the job interception feature of the BC-XBP 2.0 interface.
3. Define a Tivoli Workload Scheduler job to periodically collect the intercepted R/3 jobs.
4. Specify interception criteria in the R/3 system.
5. Specify interception criteria on the Tivoli Workload Scheduler side.

Note that jobs launched by Tivoli Workload Scheduler, or by any other external scheduler using the BC-XBP interface, cannot be intercepted. This is due to the design of the interception mechanism, which considers only jobs whose start conditions have been fulfilled due to time or SAP event restrictions.

Implementing Job Interception

This section describes what action you need to take to implement Job interception.

Activating the Job Interception Feature

Refer to SAP note 604496 to know if your R/3 system already has the BC-XBP 2.0 interface, or which SAP support package to install to enable it.

In order to enable the job interception feature, run ABAP report INITXBP2. This report shows you the current status of the job interception and parent-child features, and allows to toggle the status of both features.

Collecting Intercepted Jobs Periodically

Since the intercepted jobs remain in the scheduled status until they are re-launched, you need to define a Tivoli Workload Scheduler job to collect and restart them.

The following is the syntax of the job definition for the interception collector in Tivoli Workload Scheduler:

```
CPUNAME#JOBNAME
  SCRIPTNAME "TWShome/methods/r3batch -HIJ -c CPUNAME"
  DESCRIPTION "Collects intercepted jobs on SAP XA CPUNAME"
  STREAMLOGON TWSuser
  RECOVERY STOP
```

where:

CPUNAME

The name of the extended agent workstation.

JOBNAME

The name you want to use for the Tivoli Workload Scheduler job.

TWShome

The complete path to your Tivoli Workload Scheduler installation.

TWSuser

The name of the Tivoli Workload Scheduler user that will start the access method.

The interception collector job should run at periodical intervals; for example, every 10 min. It will retrieve all the jobs that have been intercepted since the last run of the interception collector, and relaunch them according to a template as described next.

Setting up Interception Criteria on the R/3 System

On 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 executed normally.

You can maintain the entries in this table by using transaction se16. Here you can set the following:

- Client number
- Job mask
- User mask.

Setting up Interception Criteria on Tivoli Workload Scheduler

On Tivoli Workload Scheduler, interception criteria are defined and used through the following:

- Interception criteria files
- Template files

Using Interception Criteria Files: The interception criteria for each R/3 system are stored in a separate file named *CPUNAME_r3batch.icp* and located in the *TWShome/methods/r3batch_icp* directory of the Tivoli Workload Scheduler host. This file is read by r3batch when it executes the interception collector job.

Each of the lines in the file is considered an interception criterion and is characterized by the following syntax:

```
client job_mask user_mask template
```

where *client*, *job_mask* and *user_mask* correspond to the columns of the TBCICPT1 table, while *template* is the name of the template that should be used by the

interception collector to re-launch the R/3 job. The parameters are separated by one or more spaces. You can use any text editor to maintain the contents of this file.

Using Template Files: A template is a file with extension `.jdf` located in the same directory (`TWShome/methods/r3batch_icp`) as the interception criteria file. The template file contains instructions for the interception collector about how to execute the intercepted 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.

If the user template file is empty, then a template file named `default.jdf` is utilized. If `default.jdf` does not exist, the following instructions are used:

```
alias=SAP_${RUN}_${JOBNAME}_${JOBCOUNT}
```

This means that the intercepted R/3 jobs are to be restarted immediately, due to 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 R/3 job.

Thus, the instruction set for restarting an intercepted R/3 job is retrieved following this 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 interception criteria table on R/3 contains the following entry:

```
000 * fa*
```

The interception criteria file on Tivoli Workload Scheduler contains the following entry:

```
000 * fa* at1700
```

The template file `at1700.jdf` contains the following entry:

```
alias;at=1700
```

This implies that all jobs, started in client `000` by R/3 users whose user name begins with `fa`, will be intercepted. The interception collector will restart the jobs using the instructions from the template file `at1700.jdf`. The R/3 jobs will be restarted at 17:00 with a random name, due to the `alias` command.

Using Placeholders: In the template files you can use a number of placeholders that are replaced by the interception collector at runtime. They are listed in the following table.

Table 19. Placeholders for Job Interception Template Files.

Placeholder	Description
<code>\$CPU</code>	The name of the extended agent workstation on which the interception collector runs.
<code>\$CLIENT</code>	The client number of the intercepted R/3 job.

Table 19. Placeholders for Job Interception Template Files. (continued)

Placeholder	Description
\$JOBNAME	The name of the intercepted R/3 job.
\$JOBID	The job ID of the intercepted R/3 job.
\$USER	The name of the user who launched the R/3 job.
\$JOBNUM	The job number of the interception collector.
\$RUN	The current run number of the interception collector.
\$SCHED	The schedule name of the interception collector.
\$RAND	Random number.

Thus, the template

```
alias=ICP_$RAND_$JOBNAME_$JOBID_$CLIENT;at=1000
```

would instruct the interception collector to restart the 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

The Parent-child Feature

In some situations, an R/3 job dynamically spawns a number of other jobs; for instance, to distribute the workload to the free application servers. A prominent example are the mass activity jobs of the R/3 FI-CA component. Before BC-XBP 2.0, it was difficult for external schedulers to handle this situation, because the business process is usually not over with the termination of the initial job (parent job), but with the termination of all subjobs (child jobs).

The BC-XBP 2.0 interface allows to determine whether a job has launched subjobs, together with their names and IDs, so it is now possible to track them.

To activate this feature, use the INITXBP2 ABAP report, which can also be used to toggle the status of job interception.

When the parent-child feature is active, Tivoli Workload Scheduler considers an R/3 job as finished only after all its child jobs have terminated; in other words, the status of the Tivoli Workload Scheduler job remains as EXEC as long as the parent job or any of its child jobs is running.

The status of the Tivoli Workload Scheduler job becomes SUCC if the parent job and all child jobs terminated successfully. If any of the jobs was terminated by an error, the status of the Tivoli Workload Scheduler job becomes ABEND.

Note that the parent-child feature can interfere with job interception since, although the parent job cannot be intercepted, any of its child jobs can be intercepted if it matches the interception criteria. In this case, the Tivoli Workload Scheduler job remains in the EXEC status until the intercepted child job has been re-launched and has terminated.

The joblogs of the child jobs are appended in the Tivoli Workload Scheduler stdlist after the joblog of the parent job.

Chapter 23. National Language Support

Using the new configuration files schema, with local and global configuration files, you can setup r3batch in order to use different codepages and languages for both its output and the connection with a remote R/3 system.

This 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 SAP variants.

Note: Be aware of the supported configurations and limitations of this feature.

National Language Support Options

Option TWSXA_CP affects the codepage used to open the connection between r3batch and the target R/3 system. Its value could be one of the R/3 codepages installed in the R/3 system; the default value is the SAP codepage 1100, similar to the standard ISO8859-1.

TWSXA_LANG affects the language that r3batch uses to login; it can be one of the following:

1. English (EN, the default value)
2. German (GE)
3. Japanese (JA)

TWSMETH_CP is the codepage that r3batch uses for its output. This is an advanced option and it can alter the output and the log file. Its default value is the codepage used by the Tivoli Workload Scheduler workstation that hosts r3batch. You are recommended to use this option carefully.

Note: This is true if you use feature level 1.2 of the Job Scheduling Console. Feature level 1.3 interacts with r3batch via UTF-8, therefore this limitation no longer exists.

TWSMETH_LANG affects the catalog language used by r3batch. This version contains the English message catalog only, therefore this option has no actual effect.

R/3 Supported Codepages

Tivoli Workload Scheduler for Applications for R3 is able to communicate with R/3 using the following codepages:

Table 20. R/3 Supported Codepages

R/3 Codepages	Description
1100	8859-1, the default value
1103	MS 850
8000	SJIS: Shift JIS
8300	BIG5: Traditional Chinese
8400	GBK: Simplified Chinese

Supported Configurations and Limitations

The following applies if you use feature level 1.2 of the Job Scheduling Console. Feature level 1.3 interacts with r3batch via UTF-8, therefore this limitation no longer exists.

Tivoli Workload Scheduler for Applications for R3 requires to be installed on a workstation that uses a codepage compatible with the one on the target R/3 systems. For instance, if you want to use a Japanese R/3 system that uses the SAP 8000 codepage, you should install r3batch on a workstation that uses a Japanese codepage like SJIS or MS932. If you want to retrieve the r3batch output with the Job Scheduling Console, you must run the Job Scheduling Console on a system that uses the same codepage of the Tivoli Workload Scheduler workstation where r3batch is installed. If the hosting agent uses MS932, then you should run the Job Scheduling Console on a Japanese system with that codepage.

Part 5. z/OS Access Method

Chapter 24. Introduction

IBM Tivoli Workload Scheduler for Applications for z/OS gives you the ability to schedule and control OS/390 or z/OS jobs using the sophisticated job scheduling features of Tivoli Workload Scheduler.

Note: Throughout this manual, the term z/OS is used to refer also to supported versions of OS/390. See the software requirements section.

Features

- 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 platforms.
- 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.

Software Requirements

IBM Tivoli Workload Scheduler for Applications requires the following software:

Software Requirements	Versions
IBM Tivoli Workload Scheduler	7.0 or higher.
Operating System	OS390 2.10 or higher or z/OS.
TCP/IP	One of the following: IBM 3.1 or higher; Interlink 3.1 or higher; Open Connect 2.3 or higher.
Job Scheduling Interface	<ul style="list-style-type: none">• JES2 or JES3.• OPC interface: OPC 2.3.0 or IBM Tivoli Workload Scheduler for z/OS.• CA7 interface: any currently supported version.

Installation Overview

IBM Tivoli Workload Scheduler for Applications for z/OS software consists of the z/OS access method that must reside on the Tivoli Workload Scheduler host, and of the gateway software that resides on z/OS. The gateway software can be installed both from a separate 3480 tape cartridge or from the CD-Rom of IBM Tivoli Workload Scheduler for Applications.

Before installing the access method, ensure that Tivoli Workload Scheduler is properly installed on the host workstation.

1. On the system that will be the host of the access method for z/OS, if the platform where you are installing is one of the following:
 - Compaq Tru64

- SGI Irix
- IBM Sequent Dynix

follow the instructions described in “Installing on Tier2 Platforms”.

Otherwise, follow the instructions described in Chapter 2, “Installing IBM Tivoli Workload Scheduler for Applications”, on page 5.

2. Install the gateway software on z/OS.
3. Set the required authorizations on z/OS.

Installing on Tier2 Platforms

The z/OS access method software for tier2 platforms is delivered as a tar file on CD. Follow these steps to install it:

1. Stop Tivoli Workload Scheduler on the workstation where you are installing the method.
2. Log on as **root**, and change your directory to *twshome/methods*.
3. Mount the installation CD, and restore the tar file with the following command:

```
tar xvf cd_folder/tarfile
```

where:

tar parameter	Description
<i>cd_folder</i>	The pathname of your CD drive or folder.
<i>tarfile</i>	The TAR file depending on the target platform. Can be one of the following: <ul style="list-style-type: none"> • TWS4APPS_ZOSAGENT_DYNIX.TAR for IBM Sequent Dynix • TWS4APPS_ZOSAGENT_IRIX.TAR for SGI Irix • TWS4APPS_ZOSAGENT_OSFTAR for Compaq True 64

4. Run the following command from the *twshome/methods* directory:

```
./mvsinstall.sh -user twsuser
```

Installing on z/OS

The gateway module is delivered on the following two media:

- A 3480 tape cartridge that was written in the non-IDRC (uncompressed) format.
- The CD-Rom of IBM Tivoli Workload Scheduler for Applications.

Both media contain the following two files:

- The load library.
- The samples of the two procedures.

It is up to your preference from which medium to download the files, which are identical in content.

Unloading the Files from the CD-Rom

The gateway files are stored in the ZOS directory of the product CD-Rom 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-Rom 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 datasets using the INDSN option. This will unload 11 load modules into the output library and 2 samples into the sample library.

Unloading the Files from the Tape

Modify and submit a JCL looking like the following one to unload the tape. Customize the job card and modify the next two 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 8.2.0      TO CUSTOMIZE    *
//*
//*****
//STEP01 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//INDD DD DSN=TWSX.V8R2M0.SAMPLES,
//      DISP=(OLD,PASS),UNIT=600,
//      VOL=SER=IBM001,
//      LABEL=(1,SL)
//OUTDD DD DSN=TWSX.V8R2M0.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.V8R2M0.SERVICE.APFLIB1,
//      DISP=(OLD,PASS),UNIT=600,
//      VOL=SER=IBM001,
//      LABEL=(2,SL)
//OUTDD DD DSN=TWSX.V8R2M0.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))
//*
```

Installation Overview

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 terminated for any reason, they should be stopped, not cancelled, to ensure that they shut down gracefully without impacting other programs that use the IEFU84 Exit.

Setting Authorizations on z/OS

After installing the product on z/OS, authorize the load library. This can be done by:

1. Issuing the SETPROG command from the console log. For example:

```
SETPROG APF,ADD,DSN=twxs.SERVICE.APFLIB1,  
VOL=xxxxxxx
```

where: xxxxxxx is the volume serial number where the load library is located,
or:

```
SETPROG APF,ADD,DSN=twxs.SERVICE.APFLIB1,VOL=SMS
```

which indicates a volume under control of 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).

Additional Notes

The following sections provide additional information for the functioning of the extended agent.

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 (discussed later) will dynamically chain 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(s). When EEWSpace is terminated, 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 the existing IEFU84 exit(s), which will continue to run normally.

Security

Security is enforced in several areas, usually, RACF®, Top Secret, ACF2, etc. The EEWSERVE job (discussed later) should have the ability to submit jobs that run under the userids that are supplied in the JCL to be submitted. The JCL should not contain passwords. This can be authorized using SURROGAT class resources in RACF, and the equivalents in ACF2 and Top Secret. PROPCNTL class resources in RACF should be used to prevent submitted jobs from running under the EEWSERVE userid. 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 userids (with or without passwords) can be submitted by EEWSERVE.

Note that Tivoli Workload Scheduler does not support the use of CA-7 security.

Console Security

The EEWSERVE job (discussed later) completely owns the console when displaying job streams and issuing modify commands to CA-7. It must have the authority to use an extended console. See your security administrator to ensure that this is the case. Note that CA-7 must be configured to allow users to issue modify commands through the master or extended consoles. OPERCMDS should be modified to allow for this functionality if it is not already allowed. See also “SYSTSIN Variables” for information about specifying a user ID of authority.

External security (see Security above) can be used to authorize the CA-7 user specified for EEWSERVE, since Workload Scheduler does not support providing a password with the user name.

Start Up

Follow these steps:

1. Customize and start the EEWSpace procedure (following the commented instructions contained therein) to start the extended agent Gateway Data Space. The job must be a started task and must not be cancelled. See “SYSTSIN Variables” for a description of parameter settings. EEWSpace will create the Data Space and install the IEFU84 exit. To terminate the job, use the STOP EEWSpace command from any z/OS console.

Notes:

- a. EEWSpace must be up before EEWSERVE is started.
 - b. To shutdown, stop EEWSERVE before stopping EEWSpace.
2. Customize and start the EEWSERVE procedure following the commented instructions contained therein. See “SYSTSIN Variables” for a description of parameter settings.
 3. To terminate the job, use the STOP EEWSERVE command from any z/OS console.

SYSTSIN Variables

The SYSTSIN variables are described below. You can modify the settings as required for your site configuration. The default settings are in parenthesis.

Table 21. SYSTSIN Variables.

Variable	Description
CA7INTERFACE(CONSOLE)	CONSOLE is the only supported interface for CA-7 and is the default value for this variable. Set to NO if you do not use this interface.
CA7NAME(CA7ONL)	The name of the CA-7 control region address space. Ignored if CA-7 is not used.
CA7OPTIONS(X'32')	The CA-7 release options. The default is determined from the CA-7 subsystem vector table entry and you should not change it unless directed to do so by IBM Technical Support.

Table 21. SYSTSIN Variables. (continued)

Variable	Description
CA7SPAN(9999)	<p>This variable applies exclusively to the CA7 interface. The number represents the minimum interval (in hours and minutes) after which it is possible to launch from Tivoli Workload Scheduler the same job without specifying a different SCHID. The default is intentionally set to 9999, an invalid value, so that it is possible to distinguish the case where the variable is not used. If specified, it must be a 4-digit number in the format HHMM. For example:</p> <ul style="list-style-type: none"> • CA7SPAN(0010) means 10 minutes. • CA7SPAN(0130) means 1 hour and 30 minutes. <p>Refer to “Other CA-7 Tips” on page 155 for more details.</p> <p>The minimum interval in minutes (four digits) after which it is possible to launch the same job from Tivoli Workload Scheduler without having to specify a different SCHID. Ignore if you set CA7INTERFACE equal to NO.</p>
CA7SUBSYSTEM(UC07)	The name of the CA-7 subsystem.
CA7USER(CA7CNLSL)	The userid that is used to issue CA-7 commands. Ignored if CA-7 is not used.
COMPLETIONCODE(LASTSTEP/MAXSTEP)	<p>The default JES multi-step job completion code option. LASTSTEP, the default, indicates that the completion code for a JES multi-step job is determined by the last executed step in the job. MAXSTEP indicates that the completion code is determined by the highest completion code of any executed step in the job. This variable sets the default for jobs that do not have the option overridden in the Tivoli Workload Scheduler setup for the job.</p>
DEBUG(NO)	If set to YES, it causes the gateway to output diagnostic messages. Use only in coordination with IBM Technical Support.
INTERLINKSUBSYSTEM(ACSS)	The name of the subsystem used by Interlink TCPIP stack. Ignored if Interlink is not used as TCPIP stack.

Table 21. SYSTSIN Variables. (continued)

Variable	Description
JESCMDCHR(\$)	The job command recognition character. The default is set to dollar sign (\$) for JES2 systems and to asterisk (*) for JES3 systems. This variable should be changed only if a different command recognition character is being used.
MAXWAIT(500)	The maximum amount of time, in hundredths of a second, to wait for a response to commands. Ignore if you set CA7INTERFACE equal to NO.
MCSSTORAGE(3)	The amount of storage, in megabytes, used by each extended console used by the gateway. Ignore if you set CA7INTERFACE equal to NO.
OPCINTERFACE(PIF)	PIF is the only supported interface for OPC or Tivoli Workload Scheduler for z/OS and is the default value for this variable. Set to NO if you do not use this interface.
OPCMMSGCLASS(*)	The message class for the dynamically allocated message logs used by OPC or Tivoli Workload Scheduler for z/OS. The asterisk causes the class to be set the same as EEWSERVE. Ignored if OPC or Tivoli Workload Scheduler for z/OS are not used.
OPCSUBSYSTEM(OPCS)	The subsystem name used for communications with the OPC or Tivoli Workload Scheduler for z/OS control region.
PEERADDRESS(0 0 0 0)	The default, 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 x-agent. Note: Do not include the "." (period) between the 0's in your code. The periods will show up in the display!
PORT(5000)	The TCP port number used by Tivoli Workload Scheduler and the gateway for communications. This must be the same as the value entered in the TCP Address field of the z/OS x-agent workstation definition.
PUTLINE(YES)	When set to YES, it directs trace information to DDNAME SYSTSPRT.

Table 21. SYSTSI Variables. (continued)

Variable	Description
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 Technical Support.
SVCDUMP(NO)	When set to YES, abends will cause an SVC dump. Use only in coordination with IBM Technical Support.
TCPIPSTACK(IBM)	The vendor of TCPIP stack (IBM, INTERLINK or OPENCONNECT).
TCPNAME(TCPIP)	The name of the TCPIP address space when the IBM version of TCPIP stack is used.
TERMINATOR(X'25')	The transaction termination character. Do not change the default unless directed to do so by IBM Technical 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 suit your needs. When you use CA-7, PUTLINE(YES) and WTP(NO) are required if DEBUG(YES).
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: <ul style="list-style-type: none"> • If you are using the JES2 interface, the level of the operating system does not make any difference for the product • If you are using the CA7 or Tivoli Workload Scheduler for z/OS interfaces, the support of z/OS V1R2 or later is ensured, as long as the release of the corresponding subsystem(s) support(s) it.

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 VxRyMz contains version, release, and modification levels.

Chapter 25. Set Up and Operation

Tivoli Workload Scheduler launches jobs on an extended agent for z/OS workstation. The extended 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 for z/OS workstation is a workstation definition linked to an instance of the z/OS system.

To launch a job on an extended agent for z/OS workstation, Tivoli Workload Scheduler executes the **access method**, passing it information about the job. The **access method** communicates with the instance of z/OS host and monitors the job through completion, writing job progress and status information to the job's standard list file.

This chapter describes how to:

- Create an options file to select options for a method.
- Define an extended agent for z/OS job
- Schedule an extended agent for z/OS job

See Chapter 4, "Defining Extended Agent Workstations", on page 29 to learn how to define an extended agent workstation in Tivoli Workload Scheduler.

Method Options Files

On all the supported platforms but Compaq Tru64, SGI Irix, and IBM Sequent Dynix, use the Option Editor to create the necessary options files. See Chapter 3, "Setting Options with the Option Editor", on page 25 for reference.

Method Options Files on Tier2 Platforms

On Compaq Tru64, SGI Irix, and IBM Sequent Dynix, create an options file to select options for a method. The file must have the same path name as the method and an `opts` extension. For example, create the following files on UNIX (assuming that the Tivoli Workload Scheduler home directory is `/usr/lib/tws`):

```
/usr/lib/tws/methods/mvsca7.opts
```

```
/usr/lib/tws/methods/mvsjes.opts
```

```
/usr/lib/tws/methods/mvsopc.opts
```

Or create the following files on Windows (assuming Tivoli Workload Scheduler is installed in the path `C:\WIN32APP\tws`):

```
c:\win32app\tws\methods\mvsca7.opts
```

```
c:\win32app\tws\methods\mvsjes.opts
```

```
c:\win32app\tws\methods\mvsopc.opts
```

For **mvsca7**, **mvsjes**, and **mvsopc** the options file entries are:

Options File Entries	Description
LJuser = <i>name</i>	Assigns the user name used by the access method to launch jobs. This must be a valid UNIX or Windows user who submits local jobs and is able to connect to Tivoli Workload Scheduler's z/OS gateway on the z/OS system. The default is the login parameter in the job definition.
CFuser = <i>name</i>	(Required) Assigns the user name used by the access method to check file dependencies. It can be the same as LJuser .
GSuser = <i>name</i>	(Optional) 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. The default is root .
CheckInterval = <i>min</i>	(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.
BlockTime = <i>min</i>	(Optional) Defines the amount of time, in minutes, the method will wait for a response to a status check before timing out. This value must be less than the value of CheckInterval (described above) and 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.
RetryCount = <i>count</i>	(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. See CheckInterval and BlockTime above for more information. The default is 10.

Defining Jobs in z/OS

In z/OS, jobs that are scheduled with Tivoli Workload Scheduler are defined as prescribed for JES, CA-7 and OPC or Tivoli Workload Scheduler for z/OS.

Defining z/OS Jobs in Tivoli Workload Scheduler

Tivoli Workload Scheduler job definitions are required for each z/OS job you intend to schedule and launch with Tivoli Workload Scheduler. They are defined like other Tivoli Workload Scheduler jobs and include job name, user name, special script name options, and optional recovery options.

See Chapter 5, "Defining Extended Agent Jobs", on page 35 for reference.

Task Definition Syntax for z/OS Jobs Scheduled with Tivoli Workload Scheduler

Following are descriptions of the task definition syntax (command line “scriptname”) for z/OS jobs that Tivoli Workload Scheduler schedules and launches via extended agent workstations using the **mvsjcs** (JES2/3 jobs), **mvsca7** (CA-7 jobs), or **mvsopc** (OPC or 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 **Command** field of the Task page of the Properties - Job Definition Window, if you use the Job Scheduling 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 **JOB CMD** 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 The JES job’s data set or partitioned data set name.

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. Note that there must be a space on both sides of the operator (< or =).

Example:

```
gold.apayable.cntl(apayjob1) = 0004
```

For CA-7 jobs:

The syntax is:

```
jobname [SCHID(nnn)] [CC(nnnn)          ] [...]  
        [RO(EQ|LT|GT|GE|LE|NE|#S|IG|NO)  ]  
        [SET(SKP|NTR|NDB)                 ]
```

where:

jobname

The name of the job in CA-7.

SCHID

The schedule id (a number 1-255) to be assigned to the job.

CC The condition code used with RO that indicates successful job completion.

RO The relational operator for condition code CC.

SET Sets job options to skip the next scheduled run (SKP), disable normal job triggering (NTR), or bypass database loading on completion (NDB).

For complete descriptions of the parameters, refer to your CA-7 Commands Guide.

Example:

```
ARCVJOB SCHID(203) CC(0001) RO(EQ) SET(NDB)
```

For OPC or Tivoli Workload Scheduler for z/OS jobs:

The syntax is:

```
appl [IA(yymddhhmm) | IATIME(hhmm) ] [...]
      [DEADLINE(yymddhhmm) | DEADLINETIME(hhmm) ]
      [PRIORITY(pri) ]
      [CPDEPR(Y|N|P|S) ]
```

where:

appl The name of the OPC or Tivoli Workload Scheduler for z/OS application to be inserted into the current plan.

IA The input arrival date and time in the form: *yymddhhmm* .

IATIME

The input arrival time in the form: *hhmm*.

DEADLINE

The deadline arrival date and time in the form: *yymddhhmm*.

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 your OPC or Tivoli Workload Scheduler for z/OS documentation.

Example:

```
PREFABJOB44 IA(0202181000) PRIORITY(5) CPDEPR(Y)
```

Task Definition Syntax for Other z/OS Jobs

Following are descriptions of the task definition syntax (command line “*scriptname*”) for z/OS jobs that were not launched by Tivoli Workload Scheduler, but 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 non-Tivoli Workload Scheduler jobs can be JES, CA-7 or OPC or Tivoli Workload Scheduler for z/OS.

The basic syntax is:

```
tws-job follows xagent::"mvs-job"
```

where:

tws-job The name of the Tivoli Workload Scheduler job that depends on the completion of the specified z/OS job.

xagent The name of the Tivoli Workload Scheduler x-agent workstation associated with the scheduler of the z/OS job— that is, an x-agent defined with the **mvsjes**, **mvsca7**, or **mvsopc** method. Note that the two colons are a required delimiter.

mvs-job

The identification of the z/OS job. Note that 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. Note that there must be a space on both sides of the operator (< or =).

Example:

```
job5 follows jesworkstation::"apayable = 0004"
```

For CA-7 jobs:

The syntax is:

```
"jobname [SCHID(nnn) ] [...]"
          [DATE(*|*yyddd) } ]
          [SPAN( {* } )
            {scope }
            {date }
            {(startdate,starttime,scope) }
            {(startdate,starttime,enddate,endtime) } ]
```

where:

jobname

The name of the job in CA-7.

SCHID

The schedule id (a number 1-255) of the job.

DATE A specific log date to be searched for the job.

SPAN The span of log records to be searched for the job.

For complete descriptions of the parameters, refer to your *CA-7 Commands Guide*.

For example:

```
joba follows workstation::"arcvjob schid(203) span(12)"
```

See also "Other CA-7 Tips" on page 155

For OPC or 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 OPC or 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 your OPC or Tivoli Workload Scheduler for z/OS documentation. For example:

```
joba follows opcworkstation::"PREFABJOB44 IA(0202181000) JOBNAME(PFJ3)"
```

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 xagent"#hlq1.hlq2.hlq3"
```

where:

tws-job The name of the Tivoli Workload Scheduler job that depends on the specified z/OS file.

xagent The name of the Tivoli Workload Scheduler x-agent workstation associated with the scheduler of the z/OS job— that is, an x-agent defined with the **mvsjes**, **mvsca7**, or **mvsopc** method. Note that the # sign is a required delimiter.

hlq1.hlq2.hlq3

The identification of the z/OS file in high level qualifier terms. Note that this string must be enclosed in quotation marks.

See "Checking for Files on z/OS" on page 156 for more information.

Chapter 26. Reference Information

Technical Overview

This chapter describes job states when operating on JES, CA-7, and OPC or Tivoli Workload Scheduler for z/OS in the Tivoli Workload Scheduler environment.

Operation with JES

The following applies when operating with JES.

Launching JES Jobs

To launch and monitor a JES job, Tivoli Workload Scheduler passes the name of the JCL data set or pds it wants executed 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. Since Workload Scheduler keeps track of both the job name and JES job ID, it is able to 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 can occur if a job fails for security reasons or JCL syntax problems.

JES Job States: The next table lists JES job states with respect to Tivoli Workload Scheduler.

Tivoli Workload Scheduler Job State	JES Job State	Comment
intro	na	Workload Scheduler is starting the method.
wait	Queued	
wait	na	If the job remains in this state, it may be due to a security violation in z/OS. Check the job on the z/OS system.
exec	Executing	Job is executing.
succ	Completed	Job's condition code meets the completion criteria in the Tivoli Workload Scheduler job definition.

Tivoli Workload Scheduler Job State	JES Job State	Comment
abend	Completed	Job's condition code does not meet the completion criteria in the Tivoli Workload Scheduler job definition, or a system or user abend has occurred. System abends codes, in hex, are prefixed with "S", and user abend codes, in decimal, are prefixed with "U". Both types of codes are written to the job's stdlist file.
extrn	na	Status unknown. Can occur only when checking a job that is used as a dependency.

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.

Operation with CA-7

The following applies when operating with CA-7.

Launching CA-7 Jobs

To launch and monitor a CA-7 job, Tivoli Workload Scheduler passes the job name, and optionally the CA-7 schedID, it wants executed to the z/OS Gateway. At the time of the submission request, the Gateway receives the CA-7 Job Number which is used to track all subsequent activity for the job. It is important to note that CA-7 itself must be authorized to accept commands from the console in order for this mechanism to work. See the CA-7 documentation for a discussion of the Modify and Login commands for more information on this feature.

CA-7 Job States

The next table lists CA-7 job states with respect to Tivoli Workload Scheduler.

Tivoli Workload Scheduler Job State	CA-7 Job State	Comment
intro	na	Workload Scheduler is starting the method.
wait	Request Queue or Ready Queue	The wait state may also indicate that the job moved from the Active queue to the Request queue as the result of an abend. The job remains in the Request queue, with a restart prompt, until the operator either restarts or cancels it.
exec	Active Queue	
succ	Completed	Job's condition code meets the completion criteria in the Tivoli Workload Scheduler job definition.
abend	na	Job's condition code does not meet the completion criteria in the Tivoli Workload Scheduler job definition. Also see note below.
extrn	na	Status unknown. Can occur only when checking a job that is used as a dependency. Also see note below.

Note: If a job remains in the **abend** or **extrn** states, it may be due to a security violation in z/OS. Check the job on the z/OS system.

Checking CA-7 Jobs

To check a CA-7 job that was not launched by Tivoli Workload Scheduler, the name of the job, and optionally the CA-7 schedID, is passed by Tivoli Workload Scheduler to the Gateway. To see if the job is currently being scheduled or executed by CA-7, the Gateway searches the following CA-7 queues, in this order: Request Queue, Ready Queue, and Active Queue. If the job is not found, Tivoli Workload Scheduler rechecks at a rate defined by the **bm check status** value in its Local Options file .

Other CA-7 Tips

- When using CA-7, CA7USER must also be a valid UNIX id.
- The limitation that in prior versions of the product forced users to uniquely identify occurrences when defining CA-7 jobs in Tivoli Workload Scheduler (that is, by adding the SCHID and/or SPAN operands) has now been removed thanks to the new initialization parameter CA7SPAN.

The logic behind the implementation of CA7SPAN is the following: if SPAN is specified in the Tivoli Workload Scheduler definition of a job, then CA7SPAN is ignored. If SPAN is not specified (or cannot be specified, as in the case of jobs launched by Tivoli Workload Scheduler), then CA7SPAN is taken into consideration and its value is used in the inquiries to CA7 as SPAN. This circumvents the limitation according to which each occurrence must be uniquely identified and makes it possible to launch from Tivoli Workload Scheduler the same job with the same SCHID, with the only condition that the interval between 2 submissions must be greater than the value specified with CA7SPAN.

It is important to know that, if specified, the value of CA7SPAN will be used in all the inquiries to CA7 (if they refer to jobs launched by Tivoli Workload Scheduler or if they are issued to check internetwork dependencies for jobs not launched by Tivoli Workload Scheduler) with the only exception of those jobs that have SPAN specified in their Tivoli Workload Scheduler definition.

- If you run products— such as AFOPER—that suppress console messages, make sure that they do not suppress CA-7 messages.

Operation with OPC or Tivoli Workload Scheduler for z/OS

The following applies when operating with OPC or with Tivoli Workload Scheduler for z/OS.

Launching OPC or Tivoli Workload Scheduler for z/OS Jobs

To launch and monitor an OPC or Tivoli Workload Scheduler for z/OS job, Tivoli Workload Scheduler passes the application name, and other optional parameters, it wants executed to the z/OS Gateway. If it exists in the OPC or Tivoli Workload Scheduler for z/OS database, the application is inserted in the current plan. The input arrival, deadline arrival, priority, and automatic dependency resolution parameters, if included, override any values specified in OPC or 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 OPC or Tivoli Workload Scheduler for z/OS.

OPC or Tivoli Workload Scheduler for z/OS Operation States

The next table lists OPC or Tivoli Workload Scheduler for z/OS operation states with respect to Tivoli Workload Scheduler.

Tivoli Workload Scheduler Job State	OPC or 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	na
extrn	na

OPC or Tivoli Workload Scheduler for z/OS Occurrence States

The next table lists OPC or Tivoli Workload Scheduler for z/OS operation occurrence states with respect to Tivoli Workload Scheduler.

Tivoli Workload Scheduler Job Stream State	OPC or Tivoli Workload Scheduler for z/OS Occurrence State
wait	pending
wait	undecided
exec	started
succ	complete
abend	error
abend	deleted
abend	N/A
extrn	N/A Status unknown. Can occur only when checking a job that is used as a dependency. Also see note below.

Checking OPC or Tivoli Workload Scheduler for z/OS Jobs

To check an OPC or 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 catalogued
- It is allocated

- It is not in use by another task

If the data set does not exist, Tivoli Workload Scheduler continues to wait and check for the file at a predetermined interval.

Note that 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 (e.g. xxxxx.xxxxx(-1) cannot be used).

Timing Considerations

When checking z/OS jobs that were not launched by Tivoli Workload Scheduler, certain timing issues are critical to assuring that any associated job dependencies are correctly resolved. These issues arise as the result of Tivoli Workload Scheduler's conception of a processing day. A full discussion can be found in the *Tivoli Workload Scheduler Reference Guide*. In short, Tivoli Workload Scheduler regenerates its production schedule each day at a preset time; for example, 6:00 am every morning. To maintain a working relationship between jobs being run under z/OS and jobs being run under Tivoli Workload Scheduler, some synchronization between the two environments is necessary. If, for example, an z/OS job is started outside of Tivoli Workload Scheduler, and completes before the start of Tivoli Workload Scheduler's processing day, it will not be visible to Tivoli Workload Scheduler and should not be used in a Tivoli Workload Scheduler job dependency. Tivoli Workload Scheduler makes no attempt to determine the status of jobs that completed before its start of day.

If synchronization is not taken into account, as discussed above, Tivoli Workload Scheduler may 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 an 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 abended in Console Manager (Conman) 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's standard list file
 3. Marks the job as **abend** in Console Manager (Conman) or Job Scheduling Console displays

Job standard lists can be displayed with the Console Manager (Conman) and with the Job Scheduling Console.

Gateway Messages

Informative messages are dynamically created by the z/OS gateway program. Along with condition-specific information, the messages may include:

- Timestamp in the *yymmdd hhmmss* format
- Module name
- Label within the module
- Macro name
- Reason code (register 0)
- Return code (register 15)

The z/OS gateway messages are listed below. For module names (*module*), see below.

```
EEWI01E yymmdd hhmmss module FAILED, MACRO macro NEAR LABEL label WITH RETURN
CODE = code & REASON CODE = code
EEWI02E yymmdd hhmmss module FAILED, MACRO macro NEAR LABEL label WITH RETURN
CODE = code & ERROR NUMBER = err
EEWI03I yymmdd hhmmss module ISSUED, MACRO macro NEAR LABEL label TO OBTAIN
SOCKET DESCRIPTOR = desc
EEWI04I yymmdd hhmmss module ISSUED, MACRO macro NEAR LABEL label
TO OBTAIN HOSTNAME = host
EEWI05I yymmdd hhmmss module TCPIP ADDRESS = ipaddress
EEWI06I yymmdd hhmmss module name = name
EEWI07I yymmdd hhmmss module ISSUED, MACRO macro NEAR LABEL label FOR num
CHARACTER MESSAGE = msg
EEWI08S yymmdd hhmmss module FAILED AT OFFSET offset
EEWI09E yymmdd hhmmss module FAILED, MACRO macro NEAR LABEL label WITH RETURN
CODE = code
EEWI10S yymmdd hhmmss module ABENDED WITH ASSOCIATED REASON CODE code
EEWI11I yymmdd hhmmss module CONSOLE name ACTIVATED WITH CONSOLE ID id
EEWI12I yymmdd hhmmss module SUBTASK name COMPLETED WITH COMPLETION CODE code
EEWI13I yymmdd hhmmss module CALL TO "name" NEAR LABEL label COMPLETED WITH
COMPLETION CODE code
EEWI14S yymmdd hhmmss module INVALID DATA SPACE FOUND NEAR LABEL label
EEWI15E yymmdd hhmmss module DATASET "dset" DDNAME "dd" FAILED TO OPEN
NEAR LABEL label
EEWI16I yymmdd hhmmss module SUBMITTED JOB "job" WITH JES IDENTIFICATION "id"
EEWI17I yymmdd hhmmss module PEERADDRESS = ipaddress
EEWI18E yymmdd hhmmss module INVALID TERMINATOR term
EEWI19E yymmdd hhmmss module NEAR LABEL label INVALID VALUE FOUND name = value
EEWI20E yymmdd hhmmss module NEAR LABEL label FAILED TO FIND ACTIVE SMF IEFU84
EXIT
EEWI21E yymmdd hhmmss errmsg
EEWI22I yymmdd hhmmss TASK task MODULE module ISSUED, MACRO macro NEAR
LABEL label TO CLOSE SOCKET DESCRIPTOR socket
EEWI24E yymmdd hhmmss module NEAR LABEL label
EEWI25I yymmdd hhmmss module NEAR LABEL label APPLICATION appl WAS INSERTED IN
CURRENT PLAN WITH INPUT ARRIVAL DATE AND TIME iadatetime
EEWI26I yymmdd hhmmss module NEAR LABEL label JOB jobname(jobid)
EEWI27I yymmdd hhmmss APPLICATION appl WAS INSERTED IN CURRENT
PLAN WITH INPUT ARRIVAL DATE AND TIME iadatetime
```

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

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 dataset name associated with open ACB/DCB
EEWTCP08	Find dataset 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 TCPIP provider
EEWTCP27	Concurrent server for Openconnect TCPIP stack
EEWTCP30	Child server task
EEWTCP31	Child server task for Interlink TCPIP stack
EEWTCP32	Child server task
EEWTCP84	SMF record exit for SMFEWTM
EEWTCP87	OPC interface
EEWTCP88	OPC interface post
EEWTCP89	CA7 BTI
EEWTCP90	Issue command and wait for response
EEWTCP91	Issues TERROR to format messages and issues WTOS
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 EEWTC99
EEWTCP96	ESTAE routine to issue failure message
EEWTCP97	Calls IKJEFF19 (DAIRFAIL/VSAMFAIL) to issue message
EEWTCP98	Calls IKJEFF02 to issue message from EEWTC99
EEWTCP99	Message table
EEWTCP7	Check CA7 job
EEWTCPA	Check application request handler

Module Name	Description
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 OPC application
EEWTCPL7	Launch CA-7 job request handler
EEWTCPLJ	Launch job request handler
EEWTCPLO	Insert OPC application
EEWTCPQ7	Query CA-7 job request handler
EEWTCPQJ	Query job request handler
EEWTCPQO	Query OPC 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

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. Common Serviceability

Chapter 27. Extended Agent Reference

This chapter describes the extended agent interface and provides information for programmers who need to create custom access methods.

What are Extended Agents?

Extended agents are used to extend the job scheduling functions of Tivoli Workload Scheduler to other systems and applications.

An extended agent is defined as a workstation that has a host and an access method. The host is any other workstation, except another extended agent. The access method is an IBM-supplied or user-supplied script or program that is executed by the host whenever the extended agent is referenced in the production plan. For example, to launch a job on an extended agent, the host executes the access method, passing it job details as command line options. The access method communicates with the external system or application to launch the job and return the status of the job.

Workstation Definition

Each extended agent must have a logical workstation definition. This logical workstation must be hosted by a Tivoli Workload Scheduler physical workstation, either a master, domain manager, or FTA workstation. The extended agent workstation definition references the name of the access method and the host workstation. When jobs are launched on the extended agent workstation, the access method is called and passes the job information to the external system.

Access Method Interface

The interface between Tivoli Workload Scheduler and an access method consists of information passed to the method on the command line, and of messages returned to the scheduler in **stdout**.

Method Command Line Syntax

The scheduler host runs an access method using the following command line syntax:

```
methodname -t task options -- taskstring
```

where:

methodname

Specifies the file name of the access method. All access methods must be stored in the directory: *TWSHOME*/methods

-t task Specifies the task to be performed, where *task* is one of the following:

LJ Launches a job.

MJ Manages a previously launched job. Use this option to resynchronize if a prior **LJ** task terminated unexpectedly.

CF Checks the availability of a file. Use this option to check file **opens** dependencies.

GS Gets the status of a job. Use this option to check job **follows** dependencies.

options Specifies the options associated with the task. See “Task Options” for more information.

taskstring

A string of up to 255 characters associated with the task. See “Task Options”.

Task Options

The task options are listed in the following table. An X means that the option is valid for the task.

Task	Options											Task String	
	-c	-n	-p	-r	-s	-d	-l	-o	-j	-q	-w		-S
LJ	X	X	X	X	X	X	X	X	X			X	<i>ljstring</i>
MJ	X	X	X	X	X	X	X	X	X				<i>mjstring</i>
CF	X	X	X							X			<i>cfstring</i>
GS	X	X	X	X		X					X		<i>gsstring</i>

-c *xagent,host,master*

Specifies the scheduler names of the extended agent, the host, and the master domain manager separated by commas.

-n *nodename*

Specifies the node name of the computer associated with the extended agent, if any. This is defined in the extended agent’s workstation definition **Node** field.

-p *portnumber*

Specifies the TCP port number associated with the extended agent, if any. This is defined in the extended agent’s workstation definition **TCP Address** field.

-r *currentrun,specificrun*

Specifies the current run number of the scheduler and the specific run number associated with the job separated by a comma. The current and specific run numbers might be different if the job was carried forward from an earlier run.

-s *jstream*

Specifies the name of the job’s job stream.

-d *scheddate,epoch*

Specifies the schedule date (*yymmdd*) and the epoch equivalent, separated by a comma.

-l *user* Specifies the job’s user name. This is defined in the job definition **Logon** field.

-o *stdlist*

Specifies the full path name of the job’s standard list file. Any output from the job must be written to this file.

- j *jobname,id*
Specifies the job's name and the unique identifier assigned by the scheduler, separated by a comma. The name is defined in the job definition **Job Name** field.
- q *qualifier*
Specifies the qualifier to be used in a test command issued by the method against the file.
- w *timeout*
Specifies the amount of time, in seconds, that the scheduler waits to get a reply on an external job before sending a SIGTERM signal to the access method. The default is 300.
- S *new name*
Specifies that the job is rerun using this name in place of the original job name. Within a job script, you can use the `jobinfo` command to return the job name and execute the script differently for each iteration. See the description of the `rerun` command in the *IBM Tivoli Workload Scheduler Reference Guide* for more information.
- *ljstring*
Used with the **LJ** task. The string from the **Script File** or **Command** field of the job definition.
- *mjstring*
Used with the **MJ** task. The information provided to the scheduler by the method in a **%CJ** response to an **LJ** task. Usually, this identifies the job that was launched. For example, a Unix method can provide the process identification (PID) of the job it launched, which is then sent by the scheduler as part of an **MJ** task.
- *cfstring*
Used with the **CF** task. For a file **opens** dependency, the string from the **Opens Files** field of the job stream definition.
- *gsstring*
Used with the **GS** task. Specifies the job whose status is checked. The format is as follows:

followsjob[,jobid]

where:

followsjob The string from the **Follows Sched/Job** list of the job stream definition.

jobid An optional job identifier received by the scheduler in a **%CJ** response to a previous **GS** task.

Method Response Messages

Methods return information to Tivoli Workload Scheduler in messages written to **stdout**. Each line starting with a percent sign (%) and ending with a new line is interpreted as a message. The messages have the following format:

%CJ *state* [*mjstring* | *jobid*]

%JS [*cputime*]

%RC *rc*

%UT [*errormessage*]

where:

CJ Changes the job state.

state The state to which the job is changed. All scheduler job states are valid except **hold** and **ready**. For the **GS** task, the following states are also valid:

ERROR

An error occurred.

EXTRN

Status is unknown.

mjstring

A string of up to 255 characters that the scheduler will include in any **MJ** task associated with the job. See 165.

jobid

A string of up to 64 characters that the scheduler will include in any **GS** task associated with the job. See 165.

JS [*cputime*]

Indicates successful completion of a job and provides its elapsed run time in seconds.

RC *rc* *rc* is a number that is interpreted by Tivoli Workload Scheduler as the return code of the extended agent job. The return code is taken into account only if a return code condition was specified in the definition of the extended agent job. If this is not the case, it is ignored and the successful completion of the extended agent job is indicated by the presence of message **%JS** [*cputime*]. Likewise, if the method does not send the **%RC** message, then the successful completion of the extended agent job is indicated by the presence of message **%JS** [*cputime*].

UT [*errormessage*]

Indicates that the requested task is not supported by the method. Displays a string of up to 255 characters that the scheduler will include in its error message.

Method Options File

You can use a method options file to specify special login information and other options. The scheduler reads the file, if it exists, before executing a method. If the file is modified after the scheduler is started, the changes take effect when it is stopped and restarted.

The file can contain scheduler options and any other method information. The options recognized by the scheduler are as follows:

LJuser=*username*

CFuser=*username*

GSuser=*username*

GStimeout=*seconds*

where:

LJuser=*username*

Specifies the login to use for the **LJ** and **MJ** tasks. The default is the login from the job definition.

CFuser=*username*

Specifies the login to use for the **CF** task. The default is **root** for Unix, and for Windows it is the user name of the account in which the scheduler was installed.

GSuser=*username*

Specifies the login to use for the **GS** tasks. The default is **root** for Unix, and for Windows it is the user name of the account in which the scheduler was installed.

GStimeout=*seconds*

Specifies the amount of time, in seconds, the scheduler waits for a response before killing the access method. The default is 300 seconds.

Note: If the extended agent's host is a Windows computer, these users must be defined as scheduler user objects.

The options file must have the same path name as its access method, with an **.opts** file extension. For example, the Windows path name of an options file for a method named **netmth** is:

- *TWSHome\methods\netmth.opts* (global).
- *TWSHome\methods\<CPUname>_netmth.opts* (local).

Method Execution

The following topics describe the interchange between Tivoli Workload Scheduler and an access method.

Launch Job (LJ) Task

The **LJ** task instructs the extended agent method to launch a job on an external system or application. Before running the method, Tivoli Workload Scheduler establishes an execution environment. The **LJuser** parameter is read from the method options file to determine the user account with which to run the method. If the parameter is not present or the options file does not exist, the user account specified in the **Logon** field of the job's definition is used. In addition, the following environment variables are set:

HOME

The login user's home directory.

LOGNAME

The login user's name.

PATH For Unix, it is set to **/bin:/usr/bin**. For Windows, it is set to **%SYSTEM%\SYSTEM32**.

TZ The time zone.

If the method cannot be executed, the job is placed in the **fail** state.

Once a method is running, it writes messages to its **stdout** that indicate the state of the job on the external system. The messages are summarized in the following table.

Task	Method Response	Tivoli Workload Scheduler Action
LJ and MJ	%CJ <i>state</i> [<i>mjstring</i>]	Sets job state to <i>state</i> . Includes <i>mjstring</i> in any subsequent MJ task.
	%JS [<i>cputime</i>]	Sets job state to succ .
	Exit code=non-zero	Sets job state to abend .
	%UT [<i>errormessage</i>] and Exit code=2	Sets job state to abend and displays <i>errormessage</i> .

A typical sequence consists of one or more %CJ messages indicating changes to the job state and then a %JS message before the method exits to indicate that the job ended successfully. If the job is unsuccessful, the method must exit without writing the %JS message. A method that does not support the LJ task, writes a %UT message to **stdout** and exits with an exit code of 2.

Manage Job (MJ) Task

The MJ task is used to synchronize with a previously launched job if the scheduler determines that the LJ task terminated unexpectedly. The scheduler sets up the environment in the same manner as for the LJ task and passes in the *mjstring*. See “Launch Job (LJ) Task” on page 167 for more information.

If the method locates the specified job, it responds with the same messages as an LJ task. If the method is unable to locate the job, it exits with a non-zero exit code, causing the scheduler to place the job in the **abend** state.

Killing a Job

While an LJ or MJ task is running, the method must trap a SIGTERM signal (signal 15). The signal is sent when an operator issues a **Kill** command through the scheduler console manager. Upon receiving the signal, the method must attempt to stop (**kill**) the job and then exit without writing a %JS message.

Check File (CF) Task

The CF task requests the extended agent method to check the availability of a file on the external system. Before running the method, the scheduler establishes an execution environment. The **CFuser** parameter is read from the method options file to determine the user account with which to run the method. If the parameter is not present or the options file does not exist, the **root** user is used on Unix and, on Windows, the user name of the account in which the scheduler was installed is used. If the method cannot be executed, the file **opens** dependency is marked as failed, that is, the file status is set to **NO** and any dependent job or job stream is not allowed to execute.

Once it is running, the method executes a test command, or the equivalent, against the file using the qualifier passed to it in the **-q** command line option. If the file test is true, the method exits with an exit code of zero. If the file test is false, the method exits with a non-zero exit code. This is summarized in the following table.

Task	Method Response	Tivoli Workload Scheduler Action
CF	Exit code=0	Set file state to YES .
	Exit code=non-zero	Set file state to NO .
	%UT [<i>errormessage</i>] and Exit code=2	Set file state to NO .

A method that does not support the **CF** task writes a **%UT** message to **stdout** and exits with an exit code of **2**.

Get Status (GS) Task

The **GS** task tells the extended agent's method to check the status of a job. This is necessary when another job is dependent on the successful completion of an external job. Before running the method, the **GSuser** parameter is read from the method options file to determine the user account with which to run the method. If the parameter is not present or the options file does not exist, the **root** user is used on Unix, and, on Windows, the user name of the account in which the scheduler was installed is used. If the method cannot be executed, the dependent job or job stream is not allowed to execute. If a *jobid* is available from a prior **GS** task, it is passed to the method.

The method checks the state of the specified job, and returns it in a **%CJ** message written to **stdout**. It then exits with an exit code of zero. At a rate set by the **bm check status** local option, the method is re-executed with a **GS** task until one of the following job states is returned:

- abend** The job ended abnormally.
- succ** The job completed successfully.
- cancl** The job was cancelled.
- done** The job is done, but its success or failure is not known.
- fail** The job could not be run.
- error** An error occurred in the method while checking job status.
- extrn** The job check failed or the job status could not be determined.

Note that **GStimeout** in the method options file specifies how long the scheduler will wait for a response before killing the method. See "Method Options File" on page 166 for more information.

Method responses are summarized in the following table:

Task	Method Response	Tivoli Workload Scheduler Action
GS	%CJ <i>state</i> [<i>jobid</i>]	Sets job state to <i>state</i> and includes <i>jobid</i> in any subsequent GS task.
	%UT [<i>errormessage</i>] and Exit code=2	Job state is unchanged.

A method that does not support the **GS** task writes a **%UT** message to **stdout** and exits with an exit code of **2**.

The cpuserinfo Command

The **cpuserinfo** command can be used in an access method to return information from a workstation definition. See the *IBM Tivoli Workload Scheduler Reference Guide* for complete command information.

Troubleshooting

The following topics are provided to help troubleshoot and debug extended agent and access method problems.

Job Standard List Error Messages

All output messages from an access method, except those that start with a percent sign (%), are written to the job's standard list (**stdlist**) file. For **GS** and **CF** tasks that are not associated with scheduler jobs, messages are written to the scheduler standard list file. For information regarding a problem of any kind, check these files.

Method Not Executable

If an access method cannot be executed, the following will occur:

- For **LJ** and **MJ** tasks, the job is placed in the **fail** state.
- For the **CF** task, the file dependency is unresolved and the dependent job remains in the **hold** state.
- For the **GS** task, the job dependency is unresolved and the dependent job remains in the **hold** state.

To get more information, review the standard list files (**stdlist**) for the job and for the scheduler.

Console Manager Messages

This error message is displayed if you issue a **start**, **stop**, **link** or **unlink** command for an extended agent:

```
Error executing command: Not implemented for extended agents. [2202.58]
```

This error is not displayed if an extended agent is selected as the result of using wildcard characters.

Composer and Compiler Messages

The following error messages are generated when Composer encounters invalid syntax in a workstation definition:

```
ACCESS METHOD is syntactically invalid [1116.45]
```

```
Duplicate ACCESS keyword [1116.46]
```

```
Missing or invalid ACCESS METHOD [1116.47]
```

If an extended agent is defined with an access method but without a host, the following message is displayed:

```
"Method needs a Host CPU"
```

Jobman Messages

For extended agents, error, warning, and information messages are written to Jobman's **stdlist** file.

A successful job launch generates the following message:

```
Launched job jobname for wkstation, #Jjobid for user username.
```

Failure to launch a job generates the following message:

```
Error launching jobname for wkstation: errortext
```

Failure of a check file task generates the following message:

```
Error invoking methodname for wkstation: errortext
```

Failure of a manage job task generates the following message:

Error managing *jobname* for *wkstation* using *methodname*: *errortext*

When a method sends a message to Jobman that is not recognized, the following message is generated:

Error: message invalid for *jobname*, #*jjobnumber* for *wkstation* using *methodname*.

"first 64 characters of the offending message"

Chapter 28. The Return Code Mapping Library

The return code mapping functionality provides a standard way of mapping messages into return code values. In addition, return code mapping is customizable by users. This functionality is available for the access methods for:

- Oracle eBusiness Suite
- Peoplesoft
- R/3

This functionality provides more granularity in defining the success/fail 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 termination status (successful or failed) based on a condition on the return code of the execution of the program/script of the job.
- The return code can also be provided to the recovery job that is associated in the job definition. This causes the recovery job to perform different processing based on the return code.

The Return Code Mapping File

This file enables you to customize your own return codes with respect to certain conditions that may affect a job when it runs. From there it is possible to set the success condition of the job from which it is possible for Tivoli Workload Scheduler 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.

Format

The syntax of the return code mapping file is the following:

```
[#] "pattern1" "pattern2"... "patternn" = RC value
```

Parameters

Optional comment. All lines starting with this symbol (#) are not used for mapping.

pattern*n*

Pattern strings delimited by double quotes (" and "). If you use only one pattern string, you can omit the double quotes. If the pattern string contains a double quote 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

In the following 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
```

where:

- 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 equal 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 as such and not as a wildcard, you need to 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.

Comments

Note the following facts:

- The order of the pattern lines is important as 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 access methods for PeopleSoft and for Oracle e-Business Suite, you can write return code mapping files associating the internal states listed in Table 22 and in Table 23 on page 175.

When no return code mapping files were defined, or when a string returned by the access method does not satisfy any of the matching patterns of the mapping file, the methods use the respective standard return codes listed in the same tables.

Table 22. Job States and Return Codes for the Access Method for PeopleSoft.

Psoft Job State	Psoft Return Code
"CANCEL"	1
"DELETE"	2
"ERROR"	3

Table 22. Job States and Return Codes for the Access Method for PeopleSoft. (continued)

Psoft Job State	Psoft Return Code
"HOLD"	4
"QUEUED"	5
"INITIATED"	6
"PROCESSING"	7
"CANCELLED"	8
"SUCCESS"	9
"NO SUCCESSPOSTED"	10
"POSTED"	11
"NOT POSTED"	12
"RESEND"	13
"POSTING"	14
"GENERATED"	15

Table 23. Job States and Return Codes for the Access Method for Oracle e-Business Suite.

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

Examples: The extended agent for Oracle e-Business Suite returns the TERMINATED string. The resulting return code would be the following depending on these conditions:

- No return code mapping file exists. In this case, the method sends a return code value of 15 as listed in Table 23 on page 175.

- A return code mapping file does exist and is

```
"NORMAL"=0  
"TERMINATED", "DELETED"=1
```

In this case, the method finds the matching pattern and sends a return code value of 1.

- The return code mapping file is

```
"NORMAL"=0  
"ERROR", "DELETED"=1
```

In this case, the method does not find the matching pattern and sends a return code value of 15 as listed in Table 23 on page 175.

Return Code Mapping File Names

Each method has its own file set 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:

```
<TWShome>/methods/rcm/<method_name>-<type>-<workstation>.rcm
```

Global mapping files should have a file name according to the following scheme:

```
<TWShome>/methods/rcm/<method_name>-<type>.rcm
```

For the PeopleSoft and Oracle e-Business access methods, *type* is always equal to rcmapi. For the R/3 access method, *type* is as described in the following section.

Return Code Mapping File Names for r3batch

The method for R3 uses the following global return code mapping files:

```
<TWShome>/rcm/methods/r3batch-joblog.rcm
```

Maps messages from the joblog of an R3 job into return code values. If this file is not present, the messages in the joblog are ignored.

```
<TWShome>/rcm/methods/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.

```
<TWShome>/rcm/methods/r3batch-spool.rcm
```

Maps messages in the job spool list of a SAP job into return code values. If this file is not present, the messages in the spool list are ignored.

```
<TWShome>/rcm/methods/r3batch-syslog.rcm
```

Maps messages in the syslog of a SAP system into return code values. If this file is not present, the messages in the syslog are ignored.

```
<TWShome>/rcm/methods/r3batch-msgsrc.rcm
```

Maps ABAP exceptions / 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.

Chapter 29. Configuring the Tracing Utility

All the access methods provide the possibility to configure tracing options. To do so, you can edit a number of trace properties in the `<access_method_name>.properties` configuration file located in the `methods` subdirectory of `TWSHome`.

The configurable trace properties are:

- Tracing level (MIN,MID,MAX)
- Trace file location (in terms of absolute path)

The following example is a configuration file for the R/3 access method. It is named `r3batch.properties`. The keywords discussed in this section are highlighted for ease of use:

```
r3batch.organization=IBM
r3batch.product=TWS4APPS
r3batch.component=R3BATCH
r3batch.trace.tracers.level=DEBUG_MIN
r3batch.trace.tracers.listenerNames=r3batch.trace.handlers.traceFile
r3batch.trace.handlers.traceFile.fileDir=home/maestro/methods
r3batch.trace.handlers.traceFile.appending=true
r3batch.trace.handlers.traceFile.formatterName=r3batch.trace.formatter
```

Through the `level` keyword, the tracing mechanism supports three trace levels:

DEBUG_MIN

Only fatal errors are listed. This is the default.

DEBUG_MID

Informational messages and warnings are also listed.

DEBUG_MAX

A most verbose debug output is displayed.

The properties file settings affect all the executions of the corresponding access method. To specify a different trace setting for a particular job, add the following option in the job definition:

```
-tracelvl=(1|2|3)
```

where:

- 1 = DEBUG_MIN
- 2 = DEBUG_MID
- 3 = DEBUG_MAX

For the R/3 access method only, you can mark the **Debug** check-box in the Properties-Job Definition window of the Job Scheduling Console to set the trace level to `DEBUG_MAX` for a specific job. Note that the **Trace** check-box enables the SAP RFC trace.

Traces are written in XML format. By default, trace files are stored in the `methods` directory. To change this path, you can use the `fileDir` keyword. Be sure that the new path you specify has already been created as an absolute path with writing rights. Always use slashes (or backslashes) when you specify a new path, even if you are on Windows.

The trace files give information about the method execution to the desired level of detail. The minimum trace level is always on, in order to guarantee a First-Failure Data Capture (FFDC) ability.

The value of the appending key determines whether a new trace file is appended to an older one (true) or overwrites it (false). A value of true is recommended at all times. In particular, it should not be changed to false while executing jobs, since this would cause the loss of traces.

Chapter 30. Troubleshooting

This chapter lists a number of problem items that you may run into when installing or using Tivoli Workload Scheduler for Applications or the access methods.

Table 24. Troubleshooting Items

Area	Item
Installation: Discovering Tivoli Workload Scheduler instances	<p>Problem:If you are on Unix and have multiple installations of Tivoli Workload Scheduler versions 7.0 or 8.1 that were not properly installed under a <code>/home/TWSuser</code> home directory, the installation program randomly lists only one of such instances and names it <code><unknown></code>.</p> <p>Solution:For each of these instances, create a proper home directory that points to the actual installation directory of Tivoli Workload Scheduler. Do the following:</p> <ol style="list-style-type: none"> 1. On root, create a directory named <code>/home/TWSuser</code>, that links to the directory where the product was installed. 2. In the file <code>/usr/unicon/components</code>, add an entry named <code>/home/TWSuser</code> that links to the entry for that particular instance.
Installation: Installing psagent	<p>Problem:While installing the access method for PeopleSoft feature, the following popup message may display:</p> <p>The ordinal 6654 could not be located in the dll MFC42u.dll</p> <p>This happens because an old version of the MFC42u.dll was found in the Windows system directory.</p> <p>Solution: Replacing the MFC42u.dll with a newer version to clear the problem. A newer version of the dll can generally be found in the PeopleSoft home directory.</p>
r3batch: Scheduling R/3 jobs	<p>Problem:When creating a SAP job, the following message appears while trying to view the details of an ABAP's variant:</p> <pre>AWS00101E Missing ABAP routine. J_101_REPORT_ALL_SELECTIONS please install the latest ABAP routine for Maestro!!</pre> <p>Solution:This defect is caused by an error in a SAP function module. SAP describes this problem and possible solutions in the SAP notes 0351293 and 0116354.</p>
Mixed environment: Messages in the job logs of psagent and zosagent	<p>If your scheduling environment is made up by Tivoli Workload Scheduler Version 8.1 and Tivoli Workload Scheduler for Applications Version 8.2, messages in the job logs for the access methods for z/OS and for PeopleSoft display double prefixes:</p> <ul style="list-style-type: none"> • <code>AWSxxxx EEWIyyyy <message text></code> for zosagent • <code>AWSxxxx EEWNyyyyy <message text></code> for psagent
Mixed environment: return codes of extended agent jobs	<p>If your scheduling environment is made up by Tivoli Workload Scheduler Version 8.2 and Tivoli Workload Scheduler for Applications Version 8.1.1, then what is displayed as the return code of a submitted extended agent job (through the Job Scheduling Console or conman) is in reality the exit code of that job. It should not be interpreted as the return code.</p>

Troubleshooting the Installation of the PeopleSoft Access Method

Installation of this access method requires that you specify the PeopleSoft environment in terms of a directory that depends on your version of PeopleSoft. Some of the access method executables are installed here. The next table lists the directories and the files you have to look for if you are troubleshooting your

installation.

Table 25. PeopleSoft Environment Paths.

PeopleSoft Version	Directory Name	Look for these Files
7.x	<PS_HOME>\cblbin where <PS_HOME> is the Peoplesoft file server home directory and is locally or remotely accessible from the workstation running the access method.	TVPUPSFT.exe cblrtss.dll
8.1x	<EXT_API_DIR>\bin\client\winx86_extapi where <EXT_API_DIR> is the directory where the PeopleSoft external APIs were installed.	TVPUPSFT.exe pspiadapter.dll
8.4x	<PS_HOME>\bin\client\winx86 where <PS_HOME> is the Peoplesoft file server home directory and is locally or remotely accessible from the workstation running the access method.	TVPUPSFT.exe pspiadapter.dll

Chapter 31. Installed Files

This chapter provides a listing of the files you should find on your system upon installation of Tivoli Workload Scheduler for Applications. The information is provided by platform and feature. It does not include the installed files for Java support other than the ones required for uninstalling the product.

The following table lists installed files on Windows.

Table 26. Installed files on Windows.

Feature	Files
PeopleSoft Access Method	<p>TWS_HOME\catalog\C\peoplesoft.cat TWS_HOME\methods\PeopleSoft\PTCUDCOP.bl TWS_HOME\methods\PeopleSoft\TVPUPSFT700.dms TWS_HOME\methods\PeopleSoft\TVPUPSFT750.dms TWS_HOME\methods\defs\psagent.def TWS_HOME\methods\enigma.exe TWS_HOME\methods\msvcrt.dll TWS_HOME\methods\oslayer.dll TWS_HOME\methods\psagent.exe TWS_HOME\methods\psagent.opts TWS_HOME\methods\psagent.properties</p> <p>For PeopleTools 7.0x and Merant NetExpress 2: PSHOME70CBL2\cblbin\TVPUPSFT.exe PSHOME70CBL2\cblbin\TVPUPSFT.gnt</p> <p>For PeopleTools 7.0x and Merant NetExpress 3: PSHOME70CBL3\cblbin\TVPUPSFT.exe PSHOME70CBL3\cblbin\TVPUPSFT.gnt</p> <p>PeopleTools 7.0x and MF COBOL Workbench 4.x: PSHOME70CWB4\cblbin\TVPUPSFT.exe PSHOME70CWB4\cblbin\TVPUPSFT.gnt</p> <p>For PeopleTools 7.5x and Merant NetExpress 2: PSHOME75CBL2\cblbin\TVPUPSFT.exe PSHOME75CBL2\cblbin\TVPUPSFT.gnt</p> <p>For PeopleTools 7.5x and Merant NetExpress 3: PSHOME75CBL3\cblbin\TVPUPSFT.exe PSHOME75CBL3\cblbin\TVPUPSFT.gnt</p> <p>For PeopleTools 8.1x: PSHOME81X\bin\client\winx86_extapi\TVPUPSFT.exe</p> <p>For PeopleTools 8.4x: PSHOME84X\bin\client\winx86\TVPUPSFT.exe</p>

Table 26. Installed files on Windows. (continued)

Feature	Files
R/3 Access Method	TWS_HOME\methods\K000443.TV1 TWS_HOME\methods\R000443.TV1 TWS_HOME\methods\r3batch.exe TWS_HOME\methods\r3batch.opts TWS_HOME\methods\r3batch.properties TWS_HOME\methods\r3batch_cp.map TWS_HOME\methods\libr3c32.dll TWS_HOME\methods\enigma.exe TWS_HOME\methods\msvcrt.dll TWS_HOME\methods\msvbvm60.dll TWS_HOME\methods\msvc60.dll TWS_HOME\methods\defs\r3batch.def TWS_HOME\catalog\C\r3batch.cat
z/OS Access Method	TWS_HOME\methods\mvsca7.exe TWS_HOME\methods\mvsca7.opts TWS_HOME\methods\mvsca7.properties TWS_HOME\methods\mvsjes.exe TWS_HOME\methods\mvsjes.opts TWS_HOME\methods\mvsjes.properties TWS_HOME\methods\mvsopc.exe TWS_HOME\methods\mvsopc.opts TWS_HOME\methods\mvsopc.properties TWS_HOME\methods\oslayer.dll TWS_HOME\methods\defs\mvsca7.def TWS_HOME\methods\defs\mvsjes.def TWS_HOME\methods\defs\mvsopc.def TWS_HOME\methods\msvcrt.dll TWS_HOME\catalog\C\mvs.cat
Common	TWS_HOME\methods\opted\jakarta-oro-2.0.6.jar TWS_HOME\methods\opted\opted.bat TWS_HOME\methods\opted\tws4apps.jar TWS_HOME\methods\opted\tws4appsbeans.jar TWS_HOME\methods\opted\uil.jar TWS_HOME\methods\opted\xercesImpl.jar TWS_HOME\methods\opted\xmlParserAPIs.jar TWS_HOME\methods\opted.exe
Uninstallation	TWS_HOME\methods_xaunist\uninstall.dat TWS_HOME\methods_xaunist\uninstall.jar TWS_HOME\methods_xaunist\uninstaller.exe

The following table lists installed files on Unix (Tier 1 platforms).

Table 27. Installed files on Unix (Tier 1 platforms).

Feature	Files
Oracle e-Business Suite Access Method	TWS_HOME/catalog/C/mcm.cat TWS_HOME/methods/MCMAGENT.opts TWS_HOME/methods/MCMAGENT.properties TWS_HOME/methods/OracleTMP/MCMAGENT TWS_HOME/methods/OracleTMP/MCMLJ.o TWS_HOME/methods/OracleTMP/MCMSTAT.o TWS_HOME/methods/OracleTMP/MCMTJ.o TWS_HOME/methods/OracleTMP/MCMprod.txt TWS_HOME/methods/OracleTMP/libusr.a TWS_HOME/methods/OracleTMP/mcm.mk TWS_HOME/methods/OracleTMP/mcminstall TWS_HOME/methods/OracleTMP/mcmoptions TWS_HOME/methods/defs/MCMAGENT.def TWS_HOME/methods/enigma
R/3 Access Method	TWS_HOME/catalog/C/r3batch.cat TWS_HOME/methods/K000443.TV1 TWS_HOME/methods/R000443.TV1 TWS_HOME/methods/defs/r3batch.def TWS_HOME/methods/enigma TWS_HOME/methods/r3batch TWS_HOME/methods/r3batch.opts TWS_HOME/methods/r3batch.properties TWS_HOME/methods/r3batch_cp.map
z/OS Access Method	TWS_HOME/catalog/C/mvs.cat TWS_HOME/methods/defs/mvsca7.def TWS_HOME/methods/defs/mvsjes.def TWS_HOME/methods/defs/mvsopc.def TWS_HOME/methods/mvsca7 TWS_HOME/methods/mvsca7.opts TWS_HOME/methods/mvsca7.properties TWS_HOME/methods/mvsjes TWS_HOME/methods/mvsjes.opts TWS_HOME/methods/mvsjes.properties TWS_HOME/methods/mvsopc TWS_HOME/methods/mvsopc.opts TWS_HOME/methods/mvsopc.properties
Common	TWS_HOME/methods/opted.bin TWS_HOME/methods/opted/jakarta-oro-2.0.6.jar TWS_HOME/methods/opted/opted.sh TWS_HOME/methods/opted/tws4apps.jar TWS_HOME/methods/opted/tws4appsbeans.jar TWS_HOME/methods/opted/uil.jar TWS_HOME/methods/opted/xercesImpl.jar TWS_HOME/methods/opted/xmlParserAPIs.jar
Uninstallation	TWS_HOME/methods/_xaunist/uninstall.dat TWS_HOME/methods/_xaunist/uninstall.jar TWS_HOME/methods/_xaunist/uninstaller

Part 7. Appendixes

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

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 document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

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.

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.

If you are viewing this information in softcopy form, the photographs and color illustrations might not appear.

Trademarks

IBM, the IBM logo, Tivoli, AIX, AS/400, Dynix, MVS/ESA, OS/390, RACF, RS/6000, Sequent, z/OS, and zSeries are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

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

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



Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

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

Glossary

A

Access method. An access method is an executable used by extended agents to connect and control job execution on other operating systems (for example, MVS) and applications (for example, Oracle Applications, Peoplesoft, and R/3). The access method must be specified in the workstation definition for the extended agent.

B

Batchman. Batchman is a process started at the beginning of each Tivoli Workload Scheduler processing day to launch jobs in accordance with the information in the Symphony file.

C

Calendar. A calendar is a defined object in the Tivoli Workload Scheduler database that contains a list of scheduling dates. Because it is a unique object defined in database, it can be assigned to multiple job streams. Assigning a calendar to a job stream causes that job stream to be executed on the days specified in the calendar. Note that a calendar can be used as an inclusionary or exclusionary run cycle.

Conman. Conman (console manager) is a legacy command-line application for managing the production environment. Conman performs the following tasks: start and stop production processes, alter and display schedules and jobs in the plan, and control workstation linking in a network.

Composer. Composer is a legacy command-line application for managing the definitions of your scheduling objects in the database.

D

Database. The database contains all the definitions you have created for scheduling objects (for example, jobs, job streams, resources, workstations, etc). In addition, the database holds other important information such as statistics of job and job stream execution, information on the user ID who created an object, and an object's last modified date. In contrast, the plan contains only those jobs and job streams (including dependent objects) that are scheduled for execution in today's production.

Deadline. The last moment in time that a job or job stream can begin execution. This corresponds to the Until time in legacy Maestro.

Dependency. A dependency is a prerequisite that must be satisfied before the execution of a job or job stream can proceed. The maximum number of dependencies permitted for a job or job stream is 40. The four types of dependencies used by Tivoli Workload Scheduler are follows dependencies, resource dependencies, file dependencies, and prompt dependencies.

Domain. A domain is a named group of Tivoli Workload Scheduler workstations consisting of one or more agents and a domain manager acting as the management hub. All domains have a parent domain except for the master domain.

Domain Manager. The management hub in a Tivoli Workload Scheduler domain. All communications to and from the agents in the domain are routed through the domain manager.

Duration. The time you expect the job to take to complete. In the Timeline view of jobs in the database, the duration is represented by a light blue bar at the center of the activity bar or by a light blue diamond.

E

Earliest start time. The time before which the job or job stream cannot start. The earliest start time is an estimate based on previous experiences running the job or job stream. However, the job or job stream can start after the time you specify as long as all other dependencies are satisfied. In the timeline, the start time is represented by the beginning (left edge) of the navy blue activity bar. For job instances, the start time that IBM Tivoli Workload Scheduler for z/OS calculates is represented by a light blue bar. See also "Actual start time" and "Planned start time".

Exclusionary run cycle. A run cycle that specifies the days a job stream cannot be run. Exclusionary run cycles take precedent over inclusionary run cycles.

Expanded database. Expanded databases allow longer names for database objects such as jobs, job streams, workstations, domains, and users. Expanded databases are configured using the dbexpand command or as an option during installation. Do not expand your database before understanding the implications and impact of this command.

Extended agent. Extended agents are used to integrate Tivoli Workload Scheduler's job control features with other operating systems (for example, MVS) and applications (for example, Oracle Applications, Peoplesoft, and R/3). Extended agents use scripts called access methods to communicate with external systems.

External job. A job from one job stream that is a predecessor for a job in another job stream. An external job is represented by a place holder icon in the Graph view of the job stream.

F

Fault-tolerant agent. An agent workstation in the Tivoli Workload Scheduler network capable of resolving local dependencies and launching its jobs in the absence of a domain manager.

Fence. The job fence is a master control over job execution on a workstation. The job fence is a priority level that a job or job stream's priority must exceed before it can execute. For example, setting the fence to 40 prevents jobs with priorities of 40 or less from being launched.

Final Job Stream. The FINAL job stream should be the last job stream that is executed in a production day. It contains a job that runs the script file Jnextday.

Follows dependency. A dependency where a job or job stream cannot begin execution until other jobs or job streams have completed successfully.

G

Global options. The global options are defined on the master domain manager in the globalopts file, and these options apply to all workstations in the Tivoli Workload Scheduler network. See also "Local options".

H

Host. A Workload Scheduler workstation required by extended agents. It can be any scheduler workstation except another extended agent.

I

Inclusionary Run Cycle. A run cycle that specifies the days a job stream is scheduled to run. Exclusionary run cycles take precedent over inclusionary run cycles.

Interactive jobs. A job that runs interactively on a Windows NT desktop.

Internal status. Internal status reflects the current status of jobs and job streams in the scheduler engine. Internal status is unique to Tivoli Workload Scheduler. See also Status.

Internetwork (INET) dependencies. A dependency between jobs or job streams in separate Tivoli Workload Scheduler networks. See also "Network agent".

Internetwork (INET) job / job stream. A job or job stream from a remote Tivoli Workload Scheduler network that is a predecessor to a job or job stream in the local network. An Internetwork job is represented by a place holder icon in the Graph view of the job stream. See also "Network agent".

J

Jnextday job. Pre- and post-production processing can be fully automated by scheduling the Jnextday job to run at the end of each day. A sample jnextday job is provided as *TWSHome\Jnextday*. The Jnextday job does the following: sets up the next day's processing (contained in the Symphony file), prints reports, carries forward unfinished job streams, and stops and restarts the scheduler.

Job. A job is a unit of work that is processed at a workstation. The job definition consists of a unique job name in the scheduler database along with other information necessary to run the job. When you add a job to a job stream, you can define its dependencies and its time restrictions such as the estimated start time and deadline.

Job Instance. A job scheduled for a specific run date in the plan. See also "Job".

Job status. See "Status".

Job Stream. A Job Stream consists of a list of jobs that execute as a unit (such as a weekly backup application), along with times, priorities and other dependencies that determine the exact order of job execution.

Job stream instance. A job stream that is scheduled for a specific run date in the plan. See also "Job stream".

L

Limit. Job limits provide a means of allocating a specific number of job slots into which Tivoli Workload Scheduler is allowed to launch jobs. A job limit can be set for each job stream, and for each workstation. For example, setting the workstation job limit to 25 permits the scheduler to have no more than 25 jobs executing concurrently on the workstation.

List. A list displays job scheduling objects. You must create separate lists for each job scheduling object. For each job scheduling object, there are two types of lists: one of definitions in the database and another of instances in the plan.

Local options. The local options are defined in the localopts file. Each workstation in the Tivoli Workload Scheduler network must have a localopts file. The settings in this file apply only to that workstation. See also "Global options".

M

Master Domain Manager. In a Tivoli Workload Scheduler network, the master domain manager maintains the files used to document the scheduling objects. It creates the plan at the start of each day, and performs all logging and reporting for the network.

N

Network agent. A type of extended agent used to create dependencies between jobs and job streams on separate Tivoli Workload Scheduler networks. See also "Internetwork (INET) dependency".

P

Parameter. Parameters are used to substitute values into your jobs and job streams. When using a parameter in a job script, the value is substituted at run time. In this case, the parameter must be defined on the workstation where it will be used. Parameters cannot be used when scripting extended agent jobs.

Plan. The plan contains all job scheduling activity planned for a period of one day. In Tivoli Workload Scheduler, the plan is created every 24 hours and consists of all the jobs, job streams, and dependency objects that are scheduled to execute for that day. All job streams for which you have created run cycles are automatically scheduled and included in the plan. As the production cycle progresses, the jobs and job streams in the plan are executed according to their time restrictions and other dependencies. Any jobs or job streams that do not execute successfully are rolled over into the next day's plan.

Planned Start Time. The time that the scheduler estimates a job instance will start. This estimate is based on start times of previous executions.

Predecessor. A job that must complete successfully before successor jobs can begin execution.

Priority . The scheduler has a queuing system for jobs and job streams in the plan. You can assign a priority level for each job and job stream from 0 to 101. A priority of 0 will not execute.

Prompt. Prompts can be used as dependencies for jobs and job streams. A prompt must be answered affirmatively for the dependent job or job stream to launch. There are two types of prompts: predefined and ad hoc. An ad hoc prompt is defined within the properties of a job or job stream and is unique to that job or job stream. A predefined prompt is defined in the scheduler database and can be used by any job or job stream.

R

Resource. Resources can represent either physical or logical resources on your system. Once defined in Tivoli Workload Scheduler database, they can be used as dependencies for jobs and job streams. For example, you can define a resource named "tapes" with a unit value of two. Then, define jobs that require two available tape drives as a dependency. Jobs with this dependency cannot run concurrently because each time a job is run the "tapes" resource is in use.

Run cycle. A run cycle specifies the days that a job stream is scheduled to run. In Tivoli Workload Scheduler, there are three types of run cycles you can specify for a job stream: a Simple run cycle, a Weekly run cycle, or a Calendar run cycle (commonly called a calendar). Note that each type of run cycle can be inclusionary or exclusionary. That is, each run cycle can define the days a job stream is included in the production cycle, or the days a job stream is excluded from the production cycle. When you define multiple run cycles to a job stream, and inclusionary and exclusionary run cycles specify the same days, the exclusionary run cycles take precedent.

S

Simple Run Cycle. A simple run cycle is a specific set of user-defined days a job stream is executed. A simple run cycle is defined for a specific job stream and cannot be used by multiple job streams. For more information see Run Cycle.

Status. Status reflects the current job or job stream status within the Job Scheduling Console. The Job Scheduling Console status is common to both Tivoli Workload Scheduler and IBM Tivoli Workload Scheduler for z/OS. See also Internal status.

stdlist file. A standard list file is created for each job launched by Tivoli Workload Scheduler. Standard list files contain header and trailer banners, echoed commands, errors, and warnings. These files can be used to troubleshoot problems in job execution.

Successor. A job that cannot start until all of the predecessor jobs on which it is dependent are completed successfully.

Symphony file. This file contains the scheduling information needed by the Production Control process (batchman) to execute the plan. The file is built and loaded during the pre-production phase. During the production phase, it is continually updated to indicate the current status of production processing: work completed, work in progress, work to be done. To manage production processing, the contents of the Symphony file (plan) can be displayed and altered with the Job Scheduling console.

T

Time restrictions. Time restrictions can be specified for both jobs and job streams. A time can be specified for execution to begin, or a time can be specified after which execution will not be attempted. By specifying both, you can define a window within which a job or job stream will execute. For jobs, you can also specify a repetition rate. For example, you can have Tivoli Workload Scheduler launch the same job every 30 minutes between the hours of 8:30 a.m. and 1:30 p.m.

Tree view. The view on the left side of the Job Scheduling Console that displays the scheduler server, groups of default lists, and groups of user created lists.

U

User . For Windows NT only, the user name specified in a job definition's "Logon" field must have a matching user definition. The definitions furnish the user passwords required by the scheduler to launch jobs.

Utility commands. A set of command-line executables for managing the scheduler.

W

Weekly Run Cycle. A run cycle that specifies the days of the week that a job stream is executed. For example, a job stream can be specified to execute every Monday, Wednesday, and Friday using a weekly run cycle. A weekly run cycle is defined for a specific job stream and cannot be used by multiple job streams. For more information see Run Cycle.

Wildcards. The wildcards for Tivoli Workload Scheduler are: ? Replaces one alpha character. % Replaces one numeric character. * Replaces zero or more alphanumeric characters. Wildcards are generally used to refine a search for one or more objects in the database. For example, if you want to display all workstations, you can enter the asterisk (*) wildcard. To get a listing of workstations site1 through site8, you can enter site%.

Workstation. A workstation is usually an individual computer on which jobs and job streams are executed.

They are defined in the Tivoli Workload Scheduler database as a unique object. A workstation definition is required for every computer that executes jobs or job streams in the Workload Scheduler network.

Workstation class. A workstation class is a group of workstations. Any number of workstations can be placed in a class. Job streams and jobs can be assigned to execute on a workstation class. This makes replication of a job or job stream across many workstations easy.

X

X-agent. See "Extended agent".

Index

Special characters

WORKSTATION

NAME_MCMAGENT.opts 45

A

ABAP modules 104

ABAP modules, importing 92

accessibility xii

adding, ABAP 104

adding, external program 104

adding, steps 104

agent

fault-tolerant 43

standard 43

application states for OPC 156

authorization profile, creating 89

B

BDC wait 125

BDC wait option 94

bm check status 155, 156

books

see publications ix, xi

C

CA7 jobs 149

CA7 operation overview 154

CA7INTERFACE 141

CA7NAME 141

CA7OPTIONS 141

CA7SPAN(9999) 142

CA7SUBSYSTEM 142

CA7USER 142

CFuser 148

changing password, RFC user 94

CheckInterval 148

compatibility 85

COMPLETIONCODE 142

connecting to multiple databases 67

connection to R/3, troubleshooting 93

console security 141

control file 92

control job execution 99

conventions

typeface xiii

correction and transport files 92

creating a job 100

creating a job in R/3 99

creating an open dependency 76

creating extended agent jobs 75

creating the authorization profile 89

customer support

see software support xii

D

data file 92

debug 94

DEBUG 142

defining jobs in Oracle Applications 47

defining jobs in peoplesoft 55

defining jobs in R/3 99

defining Oracle Applications jobs

in Tivoli Workload Scheduler 47

dependency on zOS jobs 150

diagnostic information 157

directory names, notation xiii

E

e-mail contact xii

encrypting user passwords 52

encrypting, R/3 user password 87

enigma 52

enigma program 87

environment variables, notation xiii

error messages 160

example opens dependencies 77

extended agent

overview 163

reference 163

workstation definition 163

external program 104

F

features 43, 55, 137

feedback about publications xii

G

gateway messages 160

H

host requirements 59

I

IEFU84 exit 140

installation 144

installation overview 85

installing ABAP modules 89

interface software 45

INTERLINKSUBSYSTEM 142

introduction 55

J

JCL to unload the tape 139

JES jobs 149

JES operation overview 153, 154

JESCMDCHR 143

job definitions 75

job execution 193

job states

Maestro and Oracle Applications 49

job states for CA7 154

job states for JES 153

job states for OPC 156

job status mapping 78

joblog 125

jobs

job states 109

L

LJuser 148

M

manuals

see publications ix, xi

mapping job status 78

MAXWAIT 143

MCMAGENT 45

MCMAGENT.opts 45

mcmagent.opts file

entry syntax 51

example 52

overview 51

MCSSTORAGE 143

messages 160

method 83

multiple Peoplesoft databases 67

N

notation

environment variables xiii

path names xiii

typeface xiii

O

occurrence states for OPC 156

online publications

accessing xi

OPC jobs 150

OPC operation overview 155

OPCINTERFACE 143

OPCMGCLASS 143

OPCSUBSYSTEM 143

opens dependency 76

opens dependency examples 77

OPERCMD5 141

options file 55

options file, R3OPTIONS 83

Oracle Applications jobs

defining 47

Oracle Applications x-agent

features 43

ordering publications xii
overview 55

P

password encryption 52
path names, notation xiii
PEERADDRESS 143
Peoplesoft databases, multiple 67
PORT 143
psagent 55, 67
psagent options 63
psagent options files 63
publications ix
 accessing online xi
 ordering xii
PUTLINE 143

Q

QLIMIT 144

R

R/3 connection 93
R/3 job states 109
R3BATCH method 83
R3OPTIONS file 83
RFC profile 89
RFC user 94
RFC user password 94

S

scheduling agent
 overview 83
Script File 150
security 140
software
 components 45
 requirements 43
software requirements 59, 60
software support
 contacting xii
start up 141
states, job 109
status mapping, jobs 78
SUBSYS 144
SVCDUMP 144
syntax, mcmagent.opts file 51
SYSTSIN variables 141

T

target host 103
TCP/IP version 137
TCPIP stack 144
TCPIPSTACK 144
TCPNAME 144
technical overview 153
TERMINATOR 144
Tivoli Software Information Center xi
trace 94
troubleshooting R/3 connection 93
typeface conventions xiii

V

variables, notation for xiii

W

WTP 144

Z

z/OS gateway
 unloading files from CD-Rom 138
 unloading files from tape 139
zOS job dependencies 150
ZOSV1R2 144



Printed in U.S.A.

SC32-1278-00

