User’s Guide for z/OS Access Method

User’s Guide for z/OS Access Method

Note

Before using this information and the product it supports, read the information in “Notices” on page 31.


This edition applies to version 8, release 1, modification 1 of IBM Tivoli Workload Scheduler for Applications for z/OS (product number 5698-WSE) and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2000, 2002. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Contents

Preface ........................................ v
  Maintenance Release .......................... v
  Who Should Read This Guide .............. v
  Prerequisite and Related Documents ...... v
  What This Guide Contains ................. v
Publications ................................ vi
  IBM Tivoli Workload Scheduler Publications . vi
  Related Publications ...................... vi
  Accessing Publications Online ........... vii
  Ordering Publications ..................... vii
  Providing Feedback about Publications ... vii
Contacting Customer Support ............... viii
Conventions Used in This Book .......... viii
  Typeface Conventions ...................... viii

Chapter 1. Introduction .................... 1
  Features .................................... 1
  Software Requirements .................... 1
  Installation ................................ 1
  Installing on Tivoli Workload Scheduler (UNIX) . 1
  Installing on Tivoli Workload Scheduler (Windows) . 2
  Installing on z/OS .......................... 2
  Additional Notes .......................... 4
  z/OS Gateway Version ..................... 8

Chapter 2. Set Up and Operation ........... 9
  Defining an Extended Agent for z/OS Workstation . 9
  Creating a Workstation with the Job Scheduling Console . 9
  Creating a Workstation from the Command Line 11
  Method Options Files ........................ 12
  Using a Method Multiple Times ............ 13
  Defining Jobs in z/OS ....................... 13
  Defining z/OS Jobs in Tivoli Workload Scheduler 13
  Defining a Job with the Job Scheduling Console 13
  Defining a Job from the Command Line .. 16
  Task Definition Syntax for z/OS Jobs Scheduled with Tivoli Workload Scheduler ... 16
  Task Definition Syntax for Other z/OS Jobs 18
  Scheduling z/OS Jobs ..................... 19

Chapter 3. Reference Information .......... 21
  Technical Overview ........................... 21
  Operation with JES .......................... 21
  Operation with CA-7 ........................ 22
  Other CA-7 Tips ............................ 23
  Operation with OPC or Tivoli Workload Scheduler for z/OS ............................. 24
  Checking for Files on z/OS ................ 25
  Timing Considerations ..................... 25
  Diagnostic Information .................... 26
  Gateway Messages ......................... 26

Notices ........................................ 31
  Trademarks ................................. 33

Glossary ....................................... 35

Index .......................................... 39
Preface

The IBM® Tivoli® Workload Scheduler for Applications: User’s Guide for z/OS provides task-oriented information on how to set up and use Tivoli Workload Scheduler to control and execute OS390 or z/OS jobs. This version of the user’s guide is a refresh of the Tivoli Workload Scheduler Extended Agent for MVS and OS/390 User’s Guide, GC32–0642–01. The version stated on the manual–8.1.1– indicates the refresh of the manual alone. The product is not changed from version 1.4.6.

Maintenance Release

This is a maintenance refresh of the corresponding IBM Tivoli Workload Scheduler for Applications Version 8.1.1 manual, form number SH19-8504-01. It contains updates for the APARS fixed with the Tivoli Workload Scheduler patches 8.1.1-WSE-FP01 and 8.1.1-WSE-FP02.

Throughout the book, the changed or new sections are marked by revision bars. The name of the fix pack or APAR causing the information change is also featured within the changed chapter, section, or paragraph. If your version of the product has not been fixed for a particular APAR, refer to the original version of the manual for the corresponding information.

Who Should Read This Guide

The target audience for this guide is system administrators and managers who set up and use Tivoli Workload Scheduler to control OS390 or z/OS jobs. Users of the guide should have some knowledge of:

- Tivoli Workload Scheduler
- The OS390 or z/OS environments

Prerequisite and Related Documents

The following related documentation:

- Tivoli Workload Scheduler Version 7.0 User’s Guide
- Tivoli Workload Scheduler Version 8.1 Reference Guide
- Tivoli Workload Scheduler Version 8.1 Job Scheduling Console User’s Guide

Provides information about using Tivoli Workload Scheduler to monitor and manage your job scheduling environment.

What This Guide Contains

This guide includes illustrations of screens and descriptions for the Job Scheduling Console. The IBM Tivoli Workload Scheduler for Applications: User’s Guide for z/OS Access Method contains the following sections:

- Chapter 1, “Introduction”
  Describes how to install the product.
- Chapter 2, “Set Up and Operation”
  Describes how to setup extended agent for z/OS workstations, jobs and job streams.
Chapter 3, “Reference Information”

Provides a technical overview of job states when working with JES, CA-7, or OPC or Tivoli Workload Scheduler for z/OS™ in the Tivoli Workload Scheduler environment.

Publications

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

IBM Tivoli Workload Scheduler Publications

This book is part of an extensive Tivoli Workload Scheduler library. These books can help you use Tivoli Workload Scheduler more effectively:

<table>
<thead>
<tr>
<th>Task</th>
<th>Publication</th>
<th>Order number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning and installing Tivoli Workload Scheduler.</td>
<td>Tivoli Workload Scheduler Planning and Installation Guide</td>
<td>SH19-4555</td>
</tr>
<tr>
<td>Using 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.</td>
<td>Tivoli Workload Scheduler Reference Guide</td>
<td>SH19-4556</td>
</tr>
<tr>
<td>Interpreting Tivoli Workload Scheduler error messages.</td>
<td>Tivoli Workload Scheduler Error Messages</td>
<td>SH19-4557</td>
</tr>
<tr>
<td>Setting up and using the Tivoli Workload Scheduler Plus module.</td>
<td>Tivoli Workload Scheduler Plus Module User’s Guide</td>
<td>SH19-4562</td>
</tr>
<tr>
<td>Tivoli Workload Scheduler last minute information.</td>
<td>Tivoli Workload Scheduler Release Notes</td>
<td>GI10-5788</td>
</tr>
</tbody>
</table>

Related Publications

The following documents also provide useful information related to the product:

<table>
<thead>
<tr>
<th>Task</th>
<th>Publication</th>
<th>Order number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using the Java GUI™ to manage scheduling objects and run scheduling tasks.</td>
<td>Tivoli Job Scheduling Console User’s Guide</td>
<td>SH19-4552</td>
</tr>
<tr>
<td>Using the Java GUI - Last minute changes</td>
<td>Tivoli Job Scheduling Console Release Notes</td>
<td>GI10-5781</td>
</tr>
</tbody>
</table>
Accessing Publications Online

Publications in the product libraries are included in PDF or HTML formats, or both, on the product CD. To access the publications using a Web browser, open the infocenter.html file, which is located in the appropriate publications directory on the product CD.

When IBM publishes an updated version of one or more online or hardcopy publications, they are posted to the Tivoli Information Center. You can access updated publications in the Tivoli Information Center from the following Customer Support Web site:

http://www.tivoli.com/support/documents/

The Tivoli Information Center contains the most recent version of the books in the product library in PDF or HTML formats, or both. Translated documents are also available for some products.

Note: If you print PDF documents on other than letter-sized paper, select the **Fit to page** check box in the Adobe Acrobat Print dialog (which is available when you click **File -> Print**) to ensure that the full dimensions of a letter-sized page are printed on the paper that you are using.

Ordering Publications

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

http://www.ibm.com/shop/publications/order

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, for a list of telephone numbers, see the following Web site: [http://www.tivoli.com/inside/store/lit_order.html](http://www.tivoli.com/inside/store/lit_order.html)

Providing Feedback about Publications

If you have comments or suggestions about Tivoli products and documentation, send an e-mail to pubs@tivoli.com or complete the customer feedback survey at the following Web site:

http://www.tivoli.com/support/survey/

Contacting Customer Support

If you have a problem with any Tivoli product, you can contact Customer Support for Tivoli products. See the **Customer Support Handbook** at the following Web site:

http://www.tivoli.com/support/handbook/

The handbook provides information about how to contact Customer 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
- What information you should gather before contacting support

Preface vii
Conventions Used in This Book

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

Typeface Conventions

The following typeface conventions are used in this book:

**Bold**

Lowercase and mixed-case commands, command options, and flags that appear within text appear like this, in bold type.

Graphical user interface elements (except for titles of windows and dialogs) and names of keys also appear like this, in bold type.

**Italic**

Variables, values you must provide, new terms, and words and phrases that are emphasized appear like this, in italic type.

**Monospace**

Commands, command options, and flags that appear on a separate line, code examples, output, and message text appear like this, in monospace type.

Names of files and directories, text strings you must type, when they appear within text, names of Java methods and classes, and HTML and XML tags also appear like this, in monospace type.
Chapter 1. Introduction

The IBM Tivoli Workload Scheduler for Applications for z/OS – formerly known as Tivoli Workload Scheduler extended agent for MVS – gives you the ability to schedule and control OS/390® or z/OS jobs using the sophisticated job scheduling features of IBM 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, including z/OS, UNIX®, Microsoft®, Windows®, HP 9000, and SAP R/3.
- 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 for z/OS requires the following software:

<table>
<thead>
<tr>
<th>Software Requirements</th>
<th>Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tivoli Workload Scheduler</td>
<td>7.0 or 8.1.</td>
</tr>
<tr>
<td>Operating System</td>
<td>OS390 2.6 and above or z/OS.</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>One of the following: IBM 3.1 or higher; Interlink 3.1 or higher; Open Connect 2.3 or higher.</td>
</tr>
<tr>
<td>Job Scheduling Interface</td>
<td>One of the following: JES2 or JES3; OPC 2.2.0 or higher; Tivoli Workload Scheduler for z/OS 8.1; CA-7 releases that support z/OS.</td>
</tr>
</tbody>
</table>

Note: The product does not support JES3 with z/OS 1.2 or higher.

Installation

IBM Tivoli Workload Scheduler for Applications for z/OS software consists of the z/OS access methods that reside on the Tivoli Workload Scheduler host, and of the gateway software that resides on z/OS. The gateway software is installed separately from a 3480 tape cartridge.

Installing on Tivoli Workload Scheduler (UNIX)

The z/OS access methods for Tivoli Workload Scheduler are delivered as tar files on CD. Follow these steps to install them:
1. Stop Tivoli Workload Scheduler on the local node where you are installing the method.

2. Log on as root, and change your directory to twshome.

3. Mount the installation CD, and restore the tar file with the following command:
   
   ```bash
   tar xvf cd_folder/ZOSAGENT/platform/MVS.TAR
   ```

   where:

<table>
<thead>
<tr>
<th>tar parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cd_folder</td>
<td>The pathname of your CD drive or folder.</td>
</tr>
<tr>
<td>platform</td>
<td>Your target platform from one of the following:</td>
</tr>
<tr>
<td></td>
<td>• AIX® for IBM</td>
</tr>
<tr>
<td></td>
<td>• DGUX for Data General UNIX</td>
</tr>
<tr>
<td></td>
<td>• HPUX for Hewlett-Packard</td>
</tr>
<tr>
<td></td>
<td>• INTEL for Intel-based UNIX</td>
</tr>
<tr>
<td></td>
<td>• MIPS for MIPS-based UNIX</td>
</tr>
<tr>
<td></td>
<td>• OSF for Compaq True 64</td>
</tr>
<tr>
<td></td>
<td>• SOLARIS for Sun Solaris</td>
</tr>
</tbody>
</table>

4. Execute customize as follows:

   ```bash
   /bin/sh customize -update -uname twuser
   ```

5. With conman, start Tivoli Workload Scheduler. After installation, you can obtain version information by using the version command. For example:

   ```bash
   version -f twshome/methods/version.info
   ```

### Installing on Tivoli Workload Scheduler (Windows)

The z/OS access methods for Tivoli Workload Scheduler are delivered on CD. Follow these steps to install them:

1. Stop Tivoli Workload Scheduler on the local node where you are installing the method.

2. Log on as a user in the local Administrators group.

3. Close any open Windows applications, including File Manager.

4. Insert the CD and copy the files from `cd_path\ZOSAGENT\I386NT` (or from `cd_path\ZOSAGENT\ALPHA` if you run Windows on Dec Alpha) to the `methods` directory of Tivoli Workload Scheduler.

5. Run the Setup program.
   a. On the Welcome screen, click Next.
   b. On the Account Information screen, enter the `twssuser` and the password and click Next.
   c. On the Setup Information screen, verify the information you entered and click Next.
   d. A message informs you that Tivoli Workload Scheduler is being customized. When the "Workload Scheduler has been installed" message appears, click OK.

6. Start Tivoli Workload Scheduler when the customization process has completed.

### Installing on z/OS

The gateway module is delivered on a 3480 tape cartridge that was written in the non-IDRC (uncompressed) format. The tape contains one file: the load library.
Unloading the Files

1. 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.

   ```
   //JOBXMVS JOB ('PIZZA',A), 'PIZZA XMVS', NOTIFY=PIZZA,
   //CLASS=A, MSGCLASS=A, MSGLEVEL=(1,1), TIME=1440, REGION=4096K
   //*
   //REST    PROC X1=,UNIT=,VOLSER=,TAPE=
   //*
   //LOADLIB EXEC PGM=IEBCOPY
   //SYSPRINT DD SYSOUT=*
   //INPUT DD DSN=XAGENT.LOADLIB,DISP=SHR,
   //UNIT=&TAPE,
   //    VOL=(,RETAIN,SER=XAGENT),
   //    LABEL=(1,SL,EXPDT=98000)
   //OUTPUT DD DSN=&X1..XAGENT.LOADLIB,UNIT=&UNIT,
   //    VOL=SER=&VOLSER,
   //    DISP=(NEW,CATLG,DELETE),
   //    SPACE=(CYL,(3,1,45))
   //*
   // PEND
   //*
   //STEP EXEC REST TAPE=600, /*3480 tape device
   //    UNIT=3390,
   //    VOLSER=OP2801,
   //    X1=PIZZA
   //LOADLIB.SYSIN DD *
   COPY I=INPUT, O=OUTPUT
   //*
   ```

2. Authorize the load library. This can be done by issuing the SETPROG command from the console log. For example:

   ```
   SETPROG APF,ADD,DSN=PIZZA.XAGENT.LOADLIB,
   VOL=xxxxxx
   ```
   
   where: xxxxxxx is the volume serial number where the load library is located.

   or:

   ```
   SETPROG APF,ADD,DSN=PIZZA.XAGENT.LOADLIB,VOL=SMS
   ```

   which indicates a volume under control of SMS.

3. Update the PROGxx member of SYS1.PARMLIB, or the authorization will be lost at your next IPL.

Installation Overview
The two z/OS program components of the Tivoli Workload Scheduler gateway are:

**TSITCP02**
This program establishes that Tivoli Workload Scheduler is tracking on the z/OS system. The program is started by the TSDSPACE job.

**TSITCP00**
This is the “gateway” program that manages TCP/IP communications between Tivoli Workload Scheduler and the z/OS system. It is started by the TSSERVER job. TSITCP00 translates 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. TSITCP02 is always started first, followed by TSITCP00. 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.

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 TSDSPACE job (discussed later) will dynamically chain to the IEFU84 exit. If the IEFU84 exit is currently being used, TSDSPACE will “front-end” the IEFU84 exit, obtain the information it requires, and then branch to the existing user exit(s). When TSDSPACE is terminated, it removes itself from the chain and restores the chain to its original status. It is important to note that TSDSPACE 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 TSSERVER 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 TSSERVER 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 TSSERVER.

Note that Tivoli Workload Scheduler does not support the use of CA-7 security.

Console Security

The TSSERVER job (discussed later) 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" on page 5 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 TSSERVER, since Workload Scheduler does not support providing a password with the user name.

Start Up

Follow these steps:

1. Create and submit the following JCL, called TSDSPACE, to start the extended agent Gateway Data Space. This job must be a started task and must not be cancelled.
//TSDSPACE PROC  MEMBER=TSPARMS
//IEFPROC EXEC  PGM=TSITCP02,
//               TIME=NOLIMIT
//STEPLIB DD  DSN=PIZZA.XAGENT.LOADLIB,DISP=SHR
//SYSTSIN DD  DSN=PIZZA.XAGENT.CNTL(&MEMBER),DISP=SHR
//SYSTSPRT DD  SYSOUT=*

See SYSTSIN Variables below for a description of the parameter settings.

TSDSPACE will create the Data Space and install the IEFU84 exit.

To terminate the job, issue a stop from any z/OS console. For example:

STOP TSDSPACE

Note: TSDSPACE must be up before TSSERVER is started. To shutdown, shutdown TSSERVER first before shutting down TSDSPACE.

2. Add TSITCP00 to the AUTHCMD NAMES section of SYS1.PARMLIB member IKJTSOxx, where xx is the member in which the change is made. Create and execute the following as a started task (APAR PQ72736):

//TSSERVER PROC  MEMBER=TSPARMS
//IEFPROC EXEC  PGM=IKJEFT01,PARM="TSITCP00",REGION=4M,
//              TIME=NOLIMIT
//STEPLIB DD  DISP=SHR,DSN=PIZZA.XAGENT.LOADLIB
//            DD  DISP=SHR,DSN=SYS1.SEZATCP
//SYSTSIN DD  DISP=SHR,DSN=PIZZA.XAGENT.CNTL(&MEMBER)
//SYSTSPRT DD  SYSOUT=*

See SYSTSIN Variables below for a description of the parameter settings.

3. To terminate the job, issue a stop from any z/OS console. For example:

STOP TSSERVER

SYSTSIN Variables
The (APAR PQ72736, PQ72167) SYSTSIN variables are described below. You can modify the settings as required for your site configuration. The default settings are in parenthesis.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA7INTERFACE(CONSOLE)</td>
<td>The CA-7 interface. Currently, CONSOLE is the only supported value for this variable and is the default.</td>
</tr>
<tr>
<td>CA7OPTIONS(X’32’)</td>
<td>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.</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| CA7SPAN(9999)                | 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:  
  • CA7SPAN(0010) means 10 minutes.  
  • CA7SPAN(0130) means 1 hour and 30 minutes.  
  Refer to "Other CA-7 Tips" on page 23 for more details. 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 the CA7 interface is not used. |
| CA7SUBSYSTEM(UC07)           | The name of the CA-7 subsystem.                                                                                                                                                                       
| CA7NAME(CA7ONL)              | The name of the CA-7 control region address space. Ignored if CA-7 is not used.                                                                                                                      
| CA7USER(CA7CNSL)             | The userid that is used to issue CA-7 commands. Ignored if CA-7 is not used.                                                                                                                       
| COMPLETIONCODE(LASTSTEP/MAXSTEP) | 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 overriden in the Tivoli Workload Scheduler setup for the job. |
| DEBUG(NO)                    | If set to YES, it causes the gateway to output diagnostic messages. Use only in coordination with IBM Technical Support.                                                                                 
<p>| INTERLINKSUBSYSTEM(ACSS)     | The name of the subsystem used by Interlink TCPIP stack. Ignored if Interlink is not used as TCPIP stack.                                                                                              |</p>
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JESCMDCHR($)</td>
<td>The job command recognition character. The default is set to dollar sign ($) for JES2 systems and to asterisk (*) for JES3 systems. This variable should be changed only if a different command recognition character is being used.</td>
</tr>
<tr>
<td>MAXWAIT(500)</td>
<td>The maximum amount of time, in hundredths of a second, to wait for a response to commands.</td>
</tr>
<tr>
<td>MCSSTORAGE(3)</td>
<td>The amount of storage, in megabytes, used by each extended console used by the gateway.</td>
</tr>
<tr>
<td>OPCMSGCLASS(*)</td>
<td>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 TSSERVER. Ignored if OPC or Tivoli Workload Scheduler for z/OS are not used.</td>
</tr>
<tr>
<td>OPCSUBSYSTEM(OPCS)</td>
<td>The subsystem name used for communications with the OPC or Tivoli Workload Scheduler for z/OS control region.</td>
</tr>
<tr>
<td>PEERADDRESS(0 0 0 0)</td>
<td>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 &quot;.&quot; (period) between the 0’s in your code. The periods will show up in the display!</td>
</tr>
<tr>
<td>PORT(5000)</td>
<td>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.</td>
</tr>
<tr>
<td>PUTLINE(YES)</td>
<td>When set to YES, it directs trace information to DDNAME SYSTSPRT.</td>
</tr>
<tr>
<td>QLIMIT(2000)</td>
<td>The maximum number of messages to be queued to an extended console.</td>
</tr>
<tr>
<td>SUBSYS(UNIS)</td>
<td>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.</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SVCDUMP(NO)</td>
<td>When set to YES, abends will cause an SVC dump. Use only in coordination with IBM Technical Support.</td>
</tr>
<tr>
<td>TCPNAME(TCPIP)</td>
<td>The name of the TCPIP address space when the IBM version of TCPIP stack is used.</td>
</tr>
<tr>
<td>TCPIPSTACK(IBM)</td>
<td>The vendor of TCPIP stack (IBM, INTERLINK or OPENCONNECT).</td>
</tr>
<tr>
<td>TERMINATOR(X’25’)</td>
<td>The transaction termination character. Do not change the default unless directed to do so by IBM Technical Support.</td>
</tr>
<tr>
<td>WTP(NO)</td>
<td>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).</td>
</tr>
</tbody>
</table>

**z/OS Gateway Version**

The version of TSSERVER (and TSDSPACE) appears in the first line of the TSSERVER log. For example:

`TSITCP10 VERSION RELEASE MODIFICATION LEVEL = VxRyMz`

where VxRyMz contains version, release, and modification levels.
Chapter 2. 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:
- Define an extended agent for z/OS workstation
- Define an extended agent for z/OS job
- Schedule an extended agent for z/OS job

Defining an Extended Agent for z/OS Workstation

Extended agent for z/OS workstation definitions are required for each z/OS entity through which Tivoli Workload Scheduler will schedule and launch jobs. They are defined in the standard manner and include the name of the x-agent’s host and the method name.

Creating a Workstation with the Job Scheduling Console

To define an extended agent for z/OS 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.
2. In the Properties window, complete the fields: Workstation Name, Node (host or IP address), and TCP port.
3. Select the relevant operating system. Use Other for the z/OS agent.
4. Enter the domain name of the host workstation.
5. Enter the Time Zone and the Description (optional).
6. In the Options area, select the workstation type.
7. Enter the access method: mvsopc, mvsca7, or mvsjes.
8. Enter the name of the hosting workstation or use the selection button.
9. Click OK to save and to close the window.

Figure 1 on page 10 shows the MVSCPU workstation definition in the "Properties - Workstation in Database" window of the Job Scheduling Console.
Figure 1. Defining a Workstation with the Job Scheduling Console

The fields of the Properties - Workstation in Database (Workstation Definition) window are described in the following table:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The Tivoli Workload Scheduler workstation name of the extended agent for z/OS. The name cannot be longer than 8 characters.</td>
</tr>
<tr>
<td>Node</td>
<td>The node name or IP address of the z/OS system. This can be the same for more than one extended agent for z/OS. (Appears in the Job Scheduling Console only)</td>
</tr>
</tbody>
</table>

IBM Tivoli Workload Scheduler for Applications: User’s Guide for z/OS Access Method
### Field | Description
---|---
**TCP Port:** | The TCP address (port number) of the z/OS gateway on the z/OS system. Enter the same value as the PORT parameter described in the SYSTSIN variable table in Chapter 1.

**Operating System:** | Select OTHER.

**Domain:** | The Tivoli Workload Scheduler domain name of the host workstation.

**Time Zone:** | Used to set the required time zone.

**Description:** | An optional free-form textual description of the workstation (up to 40 characters).

**Workstation Type:** | Select extended agent. (Job Scheduling Console only)

**Resolve Dependencies:** | Ignored.

**Full Status:** | Ignored.

**AUTO Link:** | Ignored.

**Ignore:** | Select if you want Tivoli Workload Scheduler to ignore this workstation definition.

**Server:** | Not used.

**Host:** | The Tivoli Workload Scheduler name of the hosting workstation. This is the Tivoli Workload Scheduler master, a fault-tolerant agent, or a standard agent.

**Access Method:** | The name of the z/OS access method that is executed by the Tivoli Workload Scheduler host (see Host CPU or Host above). The methods are:

- **mvsca7** Launch and monitor z/OS jobs via CA-7.
- **mvsjes** Launch and monitor z/OS jobs via JES2 or JES3.
- **mvsopc** Launch and monitor z/OS jobs via OPC or Tivoli Workload Scheduler for z/OS.

**Note:** The name is case sensitive (must be in lower case) in Unix.

---

See *Method Options Files* below for information about configuring the access methods.

### Creating a Workstation from the Command Line

Workstation definitions can also be entered in the Tivoli Workload Scheduler command line using the `composer` program. The workstation definition shown in the previous example can be entered as follows:

```bash
cpuname MVSCPU description "z/OS extended agent"
  os other
  node mvsesa36.rome.tivoli.com
tcpaddr 5000
domain masterdm
for Tivoli Workload Scheduler
  host ROCIOUS
  access mvsjes
end
```

---

Chapter 2. Set Up and Operation  11
Method Options Files

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/mvsc7.opts
/usr/lib/tws/methods/mvsjes.opts
/usr/lib/tws/methods/mvsoopc.opts
```

Or create the following files on Windows (assuming Tivoli Workload Scheduler is installed in the path C:WIN32APP\tws):

```
c:\win32app\tws\methods\mvsc7.opts
C:\win32app\tws\methods\mvsjes.opts
C:\win32app\tws\methods\mvsoopc.opts
```

For mvsc7, mvsjes, and mvsoopc the options file entries are:

<table>
<thead>
<tr>
<th>Options File Entries</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LJUser=name</td>
<td>(Required) Specifies the login to use for the LJ and MJ tasks. The default is the login from the job definition.</td>
</tr>
<tr>
<td>CFUser=name</td>
<td>(Required) Specifies the login to use for the CF task. The default is root for UNIX, and for Windows NT it is the user name of the account in which Workload Scheduler was installed.</td>
</tr>
<tr>
<td>GSUser=name</td>
<td>(Optional) Specifies the login to use for the GS tasks. The default is root for UNIX, and for Windows NT it is the user name of the account in which Workload Scheduler was installed.</td>
</tr>
<tr>
<td>CheckInterval=min</td>
<td>(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.</td>
</tr>
<tr>
<td>BlockTime=min</td>
<td>(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.</td>
</tr>
<tr>
<td>RetryCount=count</td>
<td>(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.</td>
</tr>
</tbody>
</table>
Using a Method Multiple Times

The same method can be used by multiple x-agents. However, this would mean that all the x-agents would use the same options file created for that method. To give each x-agent their own options file, make copies of the method and give each copy a unique name. Use these unique names in the corresponding CPU definitions for each x-agent. Using separate options files will allow some flexibility among multiple x-agents using the same method.

For example:

```
mvsopc mvsopc.opts<<<< LJuser = twsuser; CFuser = twsuser; GSuser = twsuser; CheckInterval = 2; BlockTime = 2; RetryCount = 10
mvs2opc mvs2opc.opts<<<< LJuser = twsuser; CFuser = twsuser; GSuser = twsuser; CheckInterval = 1.5; BlockTime = 1.5; RetryCount = 15
mvs3opc mvs3opc.opts<<<< LJuser = root; CFuser = root; GSuser = root; CheckInterval = 2; BlockTime = 2; RetryCount = 10
```

Using Multiple X-Agents with one TSSERVER Gateway

If using multiple MVS x-agents with only one TSSERVER Gateway there is a potential to create a bottleneck on the MVS side, and potentially losing information for some jobs because of timeouts. The solution is to create multiple TSSERVER gateways in the same Logical Partition (LPAR), or in a different LPAR. Specific TSPARM variables for additional TSSERVER gateways would have to be unique to each gateway (SUBSYS name and PORT number) in the same LPAR.

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.

Defining a Job with the Job Scheduling Console

To create a z/OS extended agent job with the Job Scheduling Console, follow these steps:

1. Either select the New Job icon in the top tool bar or from the scheduling engine’s navigation bar, click right and from the drop down menu, select New Job.
2. From the General window, enter the job stream name. Select the workstation.
3. Enter a description (optional), the priority and limit.
4. Enter the time restrictions, resources, and file information.

Figure 2 on page 14 shows the General page of the Job Scheduling Console Properties-Job Definition window.
The fields of this window are described next:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>The name by which the job is scheduled in Tivoli Workload Scheduler. The name can contain up to eight characters starting with a letter.</td>
</tr>
<tr>
<td>Workstation:</td>
<td>The name of the extended agent for z/OS on which the job will run.</td>
</tr>
<tr>
<td>Description:</td>
<td>An optional free-form textual description of the job (up to 40 characters).</td>
</tr>
<tr>
<td>Login:</td>
<td>The user name used by the access method to launch this job. This must be a valid UNIX or Windows user who is able to submit local jobs and to connect to the Tivoli Workload Scheduler z/OS gateway on the z/OS system. Ljuser, specified in the method options file, overrides this user name.</td>
</tr>
</tbody>
</table>

Figure 2. Defining a z/OS Job [General Page, Job Scheduling Console]
Parameters: The identification of the z/OS job to be launched or monitored. For specific information about JES, CA-7, OPC or Tivoli Workload Scheduler for z/OS, refer to “Task Definition Syntax for z/OS Jobs Scheduled with Tivoli Workload Scheduler” on page 16 and “Task Definition Syntax for Other z/OS Jobs” on page 18.

Stop Continue Rerun Recovery Options: These buttons determine the action taken if the job terminates abnormally. Stop Do not continue with the next job (Default). Continue Continue with the next job. Rerun Rerun the failed job.

Message Optional text user can add to describe the recovery job.
Recovery Job Any valid Tivoli Workload Scheduler job.
Recovery Workstation Any valid workstation where a Tivoli Workload Scheduler job exists.

Figure 3 shows the Task page of the Job Scheduling Console Properties-Job Definition window.

Figure 3. Defining a z/OS Job [Task Page, Job Scheduling Console]
The fields of this window are described next:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job Scheduling Console</td>
<td></td>
</tr>
<tr>
<td>&quot;TASK&quot; Window</td>
<td></td>
</tr>
<tr>
<td>Task:</td>
<td>The task command string used to schedule the job in z/OS.</td>
</tr>
<tr>
<td>Add Parameter ...:</td>
<td>The identification of the z/OS job to be launched or monitored.</td>
</tr>
</tbody>
</table>

For specific information about JES, CA-7, OPC or Tivoli Workload Scheduler for z/OS, refer to “Task Definition Syntax for z/OS Jobs Scheduled with Tivoli Workload Scheduler” and “Task Definition Syntax for Other z/OS Jobs” on page 18.

Defining a Job from the Command Line

Job definitions can also be entered in the Tivoli Workload Scheduler command line using the composer program. The job definition shown in the previous example can be entered as follows:

```
mvscpu#twsbr14
   streamlogon twsuser
   description "zOS job #twsbr14"
   scriptname "job14scr.tws.jcl(twsbr14)"
   recovery stop after iefbr14r
```

For information about using composer, refer to the Tivoli Workload Scheduler Reference Guide.

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 mvsjes (JES2/3 jobs), mvsca7 (CA-7 jobs), or mvsope (OPC or Tivoli Workload Scheduler for z/OS jobs) methods.

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)] [...]
```
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(yymmddhhmm)|IATIME(hHmm)  ] [...] 
       [DEADLINE(yymmddhhmm)|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: yymmddhhmm.

IATIME
   The input arrival time in the form: hhmm.

DEADLINE
   The deadline arrival date and time in the form: yymmddhhmm.

DEADLINETIME
   The deadline arrival time in the form: hhmm.

PRIORITY
   The priority (1-9) at which to run the application.

CPDEPR
   The current plan dependency resolution selection.
   Y  Add all successor and predecessor dependencies.
   N  Do not add any dependencies. (The default.)
   P  Add predecessor dependencies.
   S  Add successor dependencies.

For complete descriptions of the parameters, refer to your OPC or Tivoli Workload Scheduler for z/OS documentation.
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:

\[ \text{tws-job follows xagent::"mvs-job"} \]

where:

\[ \text{tws-job} \]  
The name of the Workload Scheduler-launched job that depends on the completion of the specified z/OS job.

\[ \text{xagent} \]  
The name of the Workload Scheduler x-agent workstation associated with the scheduler of the z/OS job— that is, an x-agent defined with the mvsjes, mvска7, or mvsopc method. Note that the two colons are a required delimiter.

\[ \text{mvs-job} \]  
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:

\[ \text{"jobname[<|= condcode]"} \]

where:

\[ \text{jobname} \]  
The name of the job in JES.

\[ \text{condcode} \]  
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:

\[ \text{"jobname [SCHID(nnn)] [...]"} \]

\[ \text{[DATE(*|*yyddd)]} \]

\[ \text{[SPAN( {*} )]} \]

\[ \{ \text{scope } \} \]

\[ \{ \text{date } \} \]

\[ \{( \text{startdate,starttime,scope} \} \]

\[ \{( \text{startdate,starttime,enddate,endtime} \} \]
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 23](##)

### For OPC or Tivoli Workload Scheduler for z/OS jobs:

The syntax is:

```
"application[IA(yymmdhhmm)|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)"
```

### Scheduling z/OS Jobs

All z/OS jobs are scheduled in the same manner as other Tivoli Workload Scheduler jobs and can include dependencies and other constraints, including Opens Files dependencies.

For information on running this task on the Job Scheduling Console, see the *Tivoli Workload Scheduler Version 7.0 User’s Guide*, or the Tivoli Workload Scheduler Version 8.1 *Job Scheduling Console User’s Guide*.

### Scheduling from the Command Line

Jobs and job streams can also be entered in the Tivoli Workload Scheduler command line using the `composer` program. The following example shows a job stream that includes the z/OS job TWSBR14:
schedule uxmlaster#glsked
  on monthend
  at 1600
:
  twsjob
  uxl#arjob
  mvscpu#twsbr14 follows twsjob,uxl#arjob
    opens mvscpu#"prod.ledger.file"
  gls follows mvscpu#twsbr14
end

For information about using composer, refer to the Tivoli Workload Scheduler Reference Guide.
Chapter 3. 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.

<table>
<thead>
<tr>
<th>Tivoli Workload Scheduler Job State</th>
<th>JES Job State</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>intro</td>
<td>na</td>
<td>Workload Scheduler is starting the method.</td>
</tr>
<tr>
<td>wait</td>
<td>Queued</td>
<td></td>
</tr>
<tr>
<td>wait</td>
<td>na</td>
<td>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.</td>
</tr>
<tr>
<td>exec</td>
<td>Executing</td>
<td>Job is executing.</td>
</tr>
<tr>
<td>succ</td>
<td>Completed</td>
<td>Job’s condition code meets the completion criteria in the Tivoli Workload Scheduler job definition.</td>
</tr>
<tr>
<td>Tivoli Workload Scheduler Job State</td>
<td>JES Job State</td>
<td>Comment</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>abend</td>
<td>Completed</td>
<td>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.</td>
</tr>
<tr>
<td>extrn</td>
<td>na</td>
<td>Status unknown. Can occur only when checking a job that is used as a dependency.</td>
</tr>
</tbody>
</table>

**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.

(APAR PQ74695) Our IEFU84 exit cannot take care of every job without dramatic impact on the performance of the entire system. Therefore, if invoked for a job, whose job name is not present in the Gateway Dataspace, this will exit without performing any action. The Gateway inserts the job name in the Gateway Dataspace for all the jobs submitted by Tivoli Workload Scheduler and, for those jobs not launched by Tivoli Workload Scheduler, it does that only when it receives the request to check them because they represent an internetwork dependency. Therefore the correct procedure, in order to have the above internetwork dependencies correctly, resolved is the following:

1. first of all, on the MVS side make sure that there are no occurrences of the predecessor job in any state; if there are in the output queue ALL THE OCCURRENCES MUST BE PURGED;
2. only at this point is it possible to perform on the distributed side the actions which will result in the request to check the job status: as a consequence of this request, the Gateway will not find any occurrence in the system and it will insert the job name in the Dataspace with an unknown jobid ready to be updated;
3. at this point on the MVS side it is possible to submit the predecessor job, and our IEFU84 exit (finding the name in the Dataspace) will update the corresponding entry with the JES jobid when invoked at job startup and with all the necessary information (to know that it has been completed and in which way it has been completed) when invoked at the end of the job. If the job was completed correctly, the Gateway will give back the information to Tivoli Workload Scheduler, and the internetwork dependency will be correctly resolved.

**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.

<table>
<thead>
<tr>
<th>Tivoli Workload Scheduler Job State</th>
<th>CA-7 Job State</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>intro</td>
<td>na</td>
<td>Workload Scheduler is starting the method.</td>
</tr>
<tr>
<td>wait</td>
<td>Request Queue or Ready Queue</td>
<td>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.</td>
</tr>
<tr>
<td>exec</td>
<td>Active Queue</td>
<td></td>
</tr>
<tr>
<td>succ</td>
<td>Completed</td>
<td>Job’s condition code meets the completion criteria in the Tivoli Workload Scheduler job definition.</td>
</tr>
<tr>
<td>abend</td>
<td>na</td>
<td>Job’s condition code does not meet the completion criteria in the Tivoli Workload Scheduler job definition. Also see note below.</td>
</tr>
<tr>
<td>extrn</td>
<td>na</td>
<td>Status unknown. Can occur only when checking a job that is used as a dependency. Also see note below.</td>
</tr>
</tbody>
</table>

**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.

(APAR PQ72167) 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.

<table>
<thead>
<tr>
<th>Tivoli Workload Scheduler Job State</th>
<th>OPC or Tivoli Workload Scheduler for z/OS Operation State</th>
</tr>
</thead>
<tbody>
<tr>
<td>wait</td>
<td>pending</td>
</tr>
<tr>
<td>wait</td>
<td>undecided</td>
</tr>
<tr>
<td>wait</td>
<td>wait</td>
</tr>
<tr>
<td>wait</td>
<td>ready</td>
</tr>
<tr>
<td>exec</td>
<td>started</td>
</tr>
<tr>
<td>succ</td>
<td>complete</td>
</tr>
<tr>
<td>abend</td>
<td>interrupted</td>
</tr>
<tr>
<td>abend</td>
<td>error</td>
</tr>
<tr>
<td>abend</td>
<td>deleted</td>
</tr>
<tr>
<td>abend</td>
<td>na</td>
</tr>
<tr>
<td>extrn</td>
<td>na</td>
</tr>
</tbody>
</table>

**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.
<table>
<thead>
<tr>
<th>Tivoli Workload Scheduler Job Stream State</th>
<th>OPC or Tivoli Workload Scheduler for z/OS Occurrence State</th>
</tr>
</thead>
<tbody>
<tr>
<td>wait</td>
<td>pending</td>
</tr>
<tr>
<td>wait</td>
<td>undecided</td>
</tr>
<tr>
<td>exec</td>
<td>started</td>
</tr>
<tr>
<td>succ</td>
<td>complete</td>
</tr>
<tr>
<td>abend</td>
<td>error</td>
</tr>
<tr>
<td>abend</td>
<td>deleted</td>
</tr>
<tr>
<td>abend</td>
<td>N/A</td>
</tr>
<tr>
<td>extrn</td>
<td>N/A Status unknown. Can occur only when checking a job that is used as a dependency. Also see note below.</td>
</tr>
</tbody>
</table>

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. `xxxx.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 include:

- 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.

TSI001E module FAILED, MACRO macro NEAR LABEL label WITH RETURN CODE = code & REASON CODE = code
TSI002E module FAILED, MACRO macro NEAR LABEL label WITH RETURN CODE = code & ERROR NUMBER = err
TSI003I module ISSUED, MACRO macro NEAR LABEL label TO OBTAIN SOCKET DESCRIPTOR = desc
TSI004I module ISSUED, MACRO macro NEAR LABEL label TO OBTAIN HOSTNAME = host
TSI005I module TCPIP ADDRESS = ipaddress
TSI006I module name = name
TSI007I module ISSUED, MACRO macro NEAR LABEL label FOR num CHARACTER MESSAGE = msg
TSI008S module FAILED AT OFFSET offset
TSI009E module FAILED, MACRO macro NEAR LABEL label WITH RETURN CODE = code
TSI010S module ABENDED WITH ASSOCIATED REASON CODE code
TSI011I module CONSOLE name ACTIVATED WITH CONSOLE ID id
TSI012I module SUBTASK name COMPLETED WITH COMPLETION CODE code
TSI013I module CALL TO "name" NEAR LABEL label COMPLETED WITH COMPLETION CODE code
TSI014S module INVALID DATA SPACE FOUND NEAR LABEL label
TSI015E module DATASET "dset" DDNAME "dd" FAILED TO OPEN NEAR LABEL label
TSI016I module SUBMITTED JOB "job" WITH JES IDENTIFICATION "id"
TSI017I module PEERADDRESS = ipaddress
TSI018E module INVALID TERMINATOR term
TSI019E module NEAR LABEL label INVALID VALUE FOUND name = value
TSI020E module NEAR LABEL label FAILED TO FIND ACTIVE SMF IEFU84 EXIT
TSI021E errmsg
TSI022I TASK task MODULE module ISSUED, MACRO macro NEAR LABEL label TO CLOSE SOCKET DESCRIPTOR socket
TSI024E module NEAR LABEL label
TSI025I module NEAR LABEL label APPLICATION appl WAS INSERTED IN CURRENT PLAN WITH INPUT ARRIVAL DATE AND TIME iadatetime
TSI026I task MODULE module ISSUED, MACRO macro NEAR LABEL label WITH RETURN CODE = code AND ERROR NUMBER = err

The module names are:

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSI011I</td>
<td>Previous SMF exit</td>
</tr>
<tr>
<td>TSI012I</td>
<td>Main processing routine</td>
</tr>
<tr>
<td>TSI013I</td>
<td>Server test vehicle</td>
</tr>
<tr>
<td>TSI014S</td>
<td>Main dataspace construction routine</td>
</tr>
<tr>
<td>TSI015E</td>
<td>Establish ESTAE routine to issue failure message</td>
</tr>
<tr>
<td>TSI016I</td>
<td>Find dataset name associated with open ACB/DCB</td>
</tr>
<tr>
<td>TSI017I</td>
<td>Find dataset name associated with DDNAME</td>
</tr>
<tr>
<td>TSI018E</td>
<td>Initialize and examine variables</td>
</tr>
<tr>
<td>TSI019E</td>
<td>Locate and initialize control blocks</td>
</tr>
<tr>
<td>TSI020E</td>
<td>Locate and initialize control blocks</td>
</tr>
<tr>
<td>TSI021E</td>
<td>Client test vehicle</td>
</tr>
<tr>
<td>TSI022I</td>
<td>Client test vehicle</td>
</tr>
<tr>
<td>TSI023I</td>
<td>Client test vehicle</td>
</tr>
<tr>
<td>TSI024E</td>
<td>Concurrent server</td>
</tr>
<tr>
<td>TSI025I</td>
<td>Concurrent server Interlink TCPIP provider</td>
</tr>
<tr>
<td>TSI026I</td>
<td>Concurrent server for Openconnect TCPIP stack</td>
</tr>
<tr>
<td>TSI027I</td>
<td>Child server task</td>
</tr>
<tr>
<td>TSI028I</td>
<td>Child server task for Interlink TCPIP stack</td>
</tr>
<tr>
<td>Module Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>TSITCP84</td>
<td>SMF record exit for SMFEWTM</td>
</tr>
<tr>
<td>TSITCP87</td>
<td>OPC interface</td>
</tr>
<tr>
<td>TSITCP88</td>
<td>OPC interface post</td>
</tr>
<tr>
<td>TSITCP89</td>
<td>CA7 BTI</td>
</tr>
<tr>
<td>TSITCP90</td>
<td>Issue command and wait for response</td>
</tr>
<tr>
<td>TSITCP91</td>
<td>Issues TERROR to format messages and issues WTOS</td>
</tr>
<tr>
<td>TSITCP92</td>
<td>Find substring within string</td>
</tr>
<tr>
<td>TSITCP93</td>
<td>Issue CA-7 command and wait for response</td>
</tr>
<tr>
<td>TSITCP94</td>
<td>STIMER exit to post ECB past in parm</td>
</tr>
<tr>
<td>TSITCP95</td>
<td>Calls IKJEFF02 to extract buffer using TSITCP99</td>
</tr>
<tr>
<td>TSITCP96</td>
<td>ESTAE routine to issue failure message</td>
</tr>
<tr>
<td>TSITCP97</td>
<td>Calls IKJEFF19 (DAIRFAIL/VSAMFAIL) to issue message</td>
</tr>
<tr>
<td>TSITCP98</td>
<td>Calls IKJEFF02 to issue message from TSITCP99</td>
</tr>
<tr>
<td>TSITCP99</td>
<td>Message table</td>
</tr>
<tr>
<td>TSITCPC7</td>
<td>Check CA7 job</td>
</tr>
<tr>
<td>TSITCPICA</td>
<td>Check application request handler</td>
</tr>
<tr>
<td>TSITCPICF</td>
<td>Check file request handler</td>
</tr>
<tr>
<td>TSITCPICJ</td>
<td>Check job request handler</td>
</tr>
<tr>
<td>TSITCPICO</td>
<td>Check operation request handler</td>
</tr>
<tr>
<td>TSITCPICV</td>
<td>Enumerate job</td>
</tr>
<tr>
<td>TSITCPED</td>
<td>String encrypt/decrypt</td>
</tr>
<tr>
<td>TSITCPEJ</td>
<td>Enumerate job request handler</td>
</tr>
<tr>
<td>TSITCPFJ</td>
<td>Free job request handler</td>
</tr>
<tr>
<td>TSITCPIA</td>
<td>Insert application</td>
</tr>
<tr>
<td>TSITCPIE</td>
<td>String initialization</td>
</tr>
<tr>
<td>TSITCPKJ</td>
<td>Kill job request handler</td>
</tr>
<tr>
<td>TSITCPKO</td>
<td>Delete OPC application</td>
</tr>
<tr>
<td>TSITCPL7</td>
<td>Launch CA-7 job request handler</td>
</tr>
<tr>
<td>TSITCPLJ</td>
<td>Launch job request handler</td>
</tr>
<tr>
<td>TSITCPLM</td>
<td>Insert OPC application</td>
</tr>
<tr>
<td>TSITCPQ7</td>
<td>Query CA-7 job request handler</td>
</tr>
<tr>
<td>TSITCPQJ</td>
<td>Query job request handler</td>
</tr>
<tr>
<td>TSITCPQO</td>
<td>Query OPC application</td>
</tr>
<tr>
<td>TSITCPQS</td>
<td>Query subsystem request handler</td>
</tr>
<tr>
<td>TSITCPRC</td>
<td>Return code</td>
</tr>
<tr>
<td>TSITCPSC</td>
<td>System abend</td>
</tr>
<tr>
<td>TSITCPSE</td>
<td>String encryption/decryption</td>
</tr>
<tr>
<td>TSITCPUC</td>
<td>User abend</td>
</tr>
<tr>
<td>TSITCPWT</td>
<td>Wait for specified amount of time</td>
</tr>
<tr>
<td>TSITCPXX</td>
<td>Invalid request handler</td>
</tr>
</tbody>
</table>
For descriptions of return codes (rt) and reason codes (re) refer to the appropriate IBM documentation for your system.

To assist in troubleshooting, be sure to obtain a system log and the JES log for the TSDSPACE and TSSERVER started tasks. These 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
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 disclaimers 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, and z/OS 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 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 precedence 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.

**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.

**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”.

**Host.** A Workload Scheduler workstation required by extended agents. It can be any scheduler workstation except another extended agent.

**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 TWS\home\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 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.
**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.

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.

**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.

**X**

**X-agent.** See “Extended agent”.

**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.
Index

A
access method
  types 11
  using multiple times 13
application states for OPC 24

B
bm check status 23, 25

C
CA7 jobs 16
CA7 operation overview 22
CA7INTERFACE 5
CA7NAME 6
CA7OPTIONS 5
CA7SPAN 6
CA7SUBSYSTEM 6
CA7USER 6
CFuser 12
CheckInterval 12
COMPLETIONCODE 6
console security 4
CPU Name 10
Customer Support vii

D
DEBUG 6
dependency on zOS jobs 18
diagnostic information 26

E
e-mail contact vii
error messages 29

F
features 1
feedback about publications vii

G
gateway messages 29

H
Host CPU 11

I
IEFU84 exit 4, 5
installation 8
INTERLINKSUBSYSTEM 6

J
JCL to unload the tape 3
JCL TSPACE 4
JES jobs 16
JES operation overview 21, 22
JESCMDCHR 7
job execution 35
job name 16
Job Name 14
job states for CA7 23
job states for JES 21
job states for OPC 24

L
LJuser 12
Logon 14

M
MAXWAIT 7
MCSSTORAGE 7
messages 29
mvscat 11
mvsjes 11
mvsopc 11

N
Node 10

O
occurrence states for OPC 24
online publications vii
OPC jobs 17
OPC operation overview 24
OPCMSCCLASS 7
OPCSUBSYSTEM 7
Operating System 11
OPERCMDS 4
ordering publications vii

P
PEERADDRESS 7
PORT 7
PUTLINE 7

Q
QLIMIT 7

R
recovery job 15
recovery message 15
recovery options 15
recovery workstation 15

S
scheduling 19
Script File 18
security 4
start up 4
SUBSYS 7
SVCDUMP 8
SYSTIN variables 5

T
TCP Address 11
TCP/IP version 1
TCPIP stack 8
TCPIPSTACK 8
TCPNAME 8
technical overview 21
TERMINATOR 8
TSPACE 4

U
unloading files 3

W
WTP 8

Z
zOS job dependencies 18
zOS jobs
  scheduling 19