Limited Fault-tolerant Agent for AS/400

Version 8.2 (Maintenance Release April 2004)
Limited Fault-tolerant Agent for AS/400

Version 8.2 (Maintenance Release April 2004)
Note
Before using this information and the product it supports, read the information in “Notices” on page 25.

Refreshed Edition (April 2004)
This edition applies to version 8, release 2, modification 0 of IBM Tivoli Workload Scheduler (program number 5698-WSH) and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 1991, 2004. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Contents

Preface ........................................ iv

Chapter 1. Installation and Configuration 1
Introduction .................................. 1
Features ...................................... 1
Software Requirements .................... 1
Before Installation ......................... 1
Installation Procedure ..................... 1
Installation Using RSTLICPGM ............ 1
Starting Netman ............................. 3
Adding an AS/400 Limited Fault-tolerant Agent .. 3
Job Scheduling Console GUI Example ........ 4
Command Line Example ..................... 5
Local Options ................................ 6
Setting Local Options ...................... 6
Netman Configuration File ................ 10
Uninstalling Tivoli Workload Scheduler ... 11

Chapter 2. Operation ......................... 13
Defining AS/400 Jobs in Tivoli Workload Scheduler 13
NOSTDLIST Keywork Usage ................. 13
Job Status in Tivoli Workload Scheduler .... 13
AS/400 Jobs Controlled by Tivoli Workload Scheduler: Normal Mode ................. 13
AS/400 Jobs Controlled by Tivoli Workload Scheduler: Extended Mode .......... 15
Scheduling AS/400 Jobs ..................... 17
Command Line Example ..................... 17

Chapter 3. Troubleshooting ................. 23
Limitations and Workarounds ............... 23

Notices ........................................ 25
Trademarks ................................... 26

Index ......................................... 27
Preface

The Tivoli® Workload Scheduler Limited Fault-tolerant Agent for AS/400® enables you to schedule, launch, and control jobs on an AS/400 using the sophisticated job scheduling tools of Tivoli Workload Scheduler

Maintenance Release

This is a maintenance refresh of the corresponding IBM Tivoli Workload Scheduler Version 8.2 manual, form number SC32–1280–00. It contains updates for the following problems:

• Internal Defects:
  – 168587
  – 168588
  – 168589
  – 168651
  – 168777
• APAR IY55408

Throughout the book, the changed or new sections are marked by revision bars. 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 Manual

This guide is intended for system managers who will install and configure Tivoli Workload Scheduler Limited Fault-tolerant Agents on AS/400, and system administrators who perform daily administration tasks.

Users of the guide should have some knowledge of:
• The Tivoli Workload Scheduler
• The AS/400 operating system

What This Manual Contains

The Tivoli Workload Scheduler Limited Fault-tolerant Agent for AS/400 contains the following chapters:

• [Chapter 1, “Installation and Configuration,” on page 1] Installation and Configuration
  Provides installation and set up instructions for the AS/400 agent.
  Describes how to define and schedule jobs to run the AS/400 agent.
• [Chapter 3, “Troubleshooting,” on page 23] Troubleshooting
  Provides Troubleshooting information.
Publications

This section lists publications in the Tivoli Workload Scheduler for z/OS 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.

Publications for Tivoli Workload Scheduler

The following manuals are available in the Tivoli Workload Scheduler library:

- **Tivoli Workload Scheduler Planning and Installation Guide, SC32-1273**
  Describes planning and installing Tivoli Workload Scheduler.

- **Tivoli Workload Scheduler Reference Guide, SC32-1274**
  Explains the Tivoli Workload Scheduler command line, understanding how extended and network agents work, and integrating Tivoli Workload Scheduler with NetView® and with Tivoli Business Systems Manager.

- **Tivoli Workload Scheduler Troubleshooting and Error Messages, SC32-1275**
  Interprets Tivoli Workload Scheduler error messages, and provides sources of information that will help you in solving problems with Tivoli Workload Scheduler.

- **Tivoli Workload Scheduler for Applications User’s Guide, SC32-1278**
  Describes installing, using, and troubleshooting for the Tivoli Workload Scheduler extended agents.

- **Tivoli Workload Scheduler for Applications Release Notes, SC32-1279**
  Provides last-minute information about the Tivoli Workload Scheduler extended agents.

- **Tivoli Workload Scheduler AS/400 Limited FTA User’s Guide, SC32-1280**
  Describes installing, configuring, and using Tivoli Workload Scheduler fault-tolerant agents on AS/400.

- **Tivoli Workload Scheduler Plus Module User’s Guide, SC32-1276**
  Explains setting up and using the Tivoli Workload Scheduler Plus module.

- **Tivoli Workload Scheduler Release Notes, SC32-1277**
  Explains late-breaking information about Tivoli Workload Scheduler.

Publications for Tivoli Workload Scheduler for z/OS

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

- **Customization and Tuning, SC32-1265**
  Describes how to customize Tivoli Workload Scheduler for z/OS.

- **Installation Guide, SC32-1264**
  Describes how to install Tivoli Workload Scheduler for z/OS.

- **Diagnosis Guide and Reference, SC32-1261**
  Provides information to help diagnose and correct possible problems when using the product.

- **General Information, SC32-1256**
  Describes the benefits of the entire Tivoli Workload Scheduler suite.

- **Getting Started, SC32-1262**
  Describes concepts and terminology of Tivoli Workload Scheduler for z/OS and provides instructions to get you working with the product as soon as possible.
• **IBM Job Scheduling Console Release Notes, SC32-1258**  
  Provides information about working with Tivoli Workload Scheduler, regardless of platform, from a common GUI.

• **IBM Job Scheduling Console User’s Guide, SC32-1257**  
  Provides information about working with Tivoli Workload Scheduler, regardless of platform, from a common GUI.

• **Licensed Program Specifications, GI11-4208**  
  Provides planning information to plan about Tivoli Workload Scheduler for z/OS.

• **Managing the Workload, SC32-1263**  
  Explains how to plan and schedule the workload and how to control and monitor the current plan.

• **Memo to Users, GI11-4209**  
  Provides a summary of changes for the current release of the product.

• **Messages and Codes, SC32-1267**  
  Explains messages and codes.

• **Program Directory, GI11-4203**  
  Provided with the Tivoli Workload Scheduler for z/OS installation tape, describes all of the installation materials and gives installation instructions specific to the product release level or feature number.

• **Programming Interfaces, SC32-1266**  
  Provides information to write application programs.

• **Quick Reference, SC32-1268**  
  Provides a quick and easy consultation reference to operate the product.

### Accessing Publications Online

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

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


Click the Tivoli Workload Scheduler for z/OS link to access the product library.

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

### Ordering Publications

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

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

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

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

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

**Providing Feedback about Publications**

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

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

**Accessibility**

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

For additional information, see the Accessibility Appendix in the *Tivoli Workload Scheduler for z/OS Installation Guide*.

**Contacting IBM Software Support**

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

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

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

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

**Conventions Used in This Manual**

The manual uses several typeface conventions for special terms and actions. These conventions have the following meanings:

<table>
<thead>
<tr>
<th>Information type</th>
<th>Style convention</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commands</td>
<td>All capital letters</td>
<td>CREATE</td>
</tr>
<tr>
<td>References in the text to fields on panels</td>
<td>All capital letters</td>
<td>QUANTITY</td>
</tr>
<tr>
<td>Input you should type in panel fields</td>
<td>Monospace</td>
<td>MYAPPLICATION</td>
</tr>
<tr>
<td>First time new term introduced</td>
<td>Italic</td>
<td>Application</td>
</tr>
</tbody>
</table>
**Typeface conventions**

This guide uses the following typeface conventions:

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

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

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

**Operating system-dependent variables and paths**

This guide uses the UNIX convention for specifying environment variables and for directory notation.

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

**Note:** If you are using the bash shell on a Windows system, you can use the UNIX conventions.
Limited Fault-tolerant Agent for AS/400
Chapter 1. Installation and Configuration

Introduction

The Tivoli Workload Scheduler Limited Fault-tolerant Agent for AS/400 enables you to schedule, launch, and control jobs on an AS/400 using the sophisticated job scheduling tools of Tivoli Workload Scheduler.

Features

- Use standard Tivoli Workload Scheduler job dependencies on AS/400 jobs.
- Schedule AS/400 jobs on specific days and at specific times of day.
- Prioritize AS/400 jobs to have them executed in a prescribed order.
- Define inter-dependencies between AS/400 jobs and jobs running on other systems, including UNIX®, Windows NT®, and SAP R/3.

Software Requirements

To install and use the Tivoli Workload Scheduler AS/400 Limited Fault-tolerant Agent, you must be running a supported version of OS/400®. Call Customer® Support for up-to-date information on supported versions of OS/400.

Before Installation

To install the Tivoli Workload Scheduler AS/400 Limited Fault-tolerant Agent, you must be QSECOFR or have *ALLOBJ authority. In addition, the QALWOBJRST system value must be set to *ALWPGMADP or *ALL.

Installation Procedure

Use the following procedure to install the Tivoli Workload Scheduler AS/400 Limited Fault-tolerant Agent.

Read through the instructions carefully before starting the installation.

Note: Uninstall the product before attempting to install the software over a previous installation. See "Uninstalling Tivoli Workload Scheduler" on page 11 for more information.

Installation Using RSTLICPGM

Use the following procedure to install the Tivoli Workload Scheduler AS/400 Limited Fault-tolerant Agent using the RSTLICPGM command on the AS/400.

1. Insert CD 2 of IBM Tivoli Workload Scheduler Version 8.2 in your UNIX or Windows system. On UNIX, mount the CD using the appropriate platform command.

2. Optionally, open a shell CLI for UNIX or a DOS Command Prompt on Windows.

3. Optionally, create a temporary directory to copy the product file. For example: mkdir twslftatmp

4. Go to the temporary directory. For example:
5. Copy the LFTA product file into the temporary directory. For example,

   • On UNIX:
     
     ```
     cp /MOUNT_DIR/As400/QTWS.V1R3M0.SAVF .
     ```

     where `MOUNT_DIR` is the directory under which the CD ROM was mounted.

   • On Windows:
     
     ```
     copy X:\As400\QTWS.V1R3M0.SAVF .
     ```

     where `X` is the CD ROM letter identifier.

6. On the AS/400, create a save file. For example:

   ```
   CRTSAVF QGPL/QFTASAVF
   ```

   where `QGPL/QFTASAVF` is the name of the save file.

7. On the UNIX computer, use `ftp` to copy the save file to the AS/400 save file. For example:

   ```
   ftp 124.56.34.130
   User: QSECOFR
   Password: xxxxx
   ftp> bin
   ftp> put QTWS.V1R3MO.SAVF QGPL/QFTASAVF
   ftp> quit
   ```

   where `QTWS.V1R3MO.SAVF` is the name of product file for AS/400 RISC machines.

8. On the AS/400, run the `RSTLICPGM` command. For example:

   ```
   RSTLICPGM LICPGM(5G97WKB) DEV(*SAVF) SAVF(QGPL/QFTASAVF)
   ```

9. On the AS/400, run the `CUSTOMIZE` command.

   a. To perform a new installation, use the following syntax:

   ```
   CUSTOMIZE TYPE(*NEW) THISCPU(name) MASTER(name)
   GROUP(group) COMPANY(name) EXPANDED(*YES | *NO)
   ```

   where:

   **TYPE(*NEW)**
   
   Specifies that this is a new installation of the Tivoli Workload Scheduler AS/400 Limited Fault-tolerant Agent. This parameter is required for new installations.

   **THISCPU(name)**
   
   Specifies the Tivoli Workload Scheduler name of the AS/400 fault-tolerant agent. This name must be used later to formally define the workstation in Tivoli Workload Scheduler. This parameter is required.

   **MASTER(name)**
   
   Specifies the Tivoli Workload Scheduler name of the master workstation. This parameter is required.

   **GROUP(group)**
   
   Specifies the name of the product group in which to install the AS/400 fault-tolerant agent. Product groups permit the installation of multiple instances of a product. For example, a test version of the product can be installed in a group named `TEST` and a production version can be installed in a group named `PROD`. This parameter is required.

   **COMPANY(name)**
   
   Specifies the name of your company. This parameter is required.
EXPANDED *YES

The database mode for the AS/400 Limited Fault-tolerant Agent must match the database mode for the Tivoli Workload Scheduler Master workstation. This parameter is required.

For example:
CUSTOMIZE TYPE(*NEW) THISCPU('AS400-2') MASTER(CENTRAL) GROUP(PRODUCTION) COMPANY('AJAX FOODS') EXPANDED(*YES)

b. To update an existing installation, use the following syntax:
CUSTOMIZE TYPE(*UPDATE) GROUP(group) COMPANY(name)

where:

TYPE(*UPDATE)

Specifies that this is an update of an existing installation of the Tivoli Workload Scheduler AS/400 Limited Fault-tolerant Agent. This parameter is required for update installations.

GROUP(group)

Specifies the name of the product group in to be updated. This parameter is required.

COMPANY(name)

Specifies a new company name to replace the existing name. This parameter is required.

For example:
CUSTOMIZE TYPE(*UPDATE) GROUP(PRODUCTION) COMPANY(TIVOLI)

10. To install the Tivoli Workload Scheduler unsupported utilities in the QTWSTOOLS library, enter the following command:
RSTLIB SAVLIB(QTWSTOOLS) DEV(*SAVF) SAVF(QTWS/QTWSTOOLS)

For information about the utilities, refer to "Unsupported Tivoli Workload Scheduler Utilities" on page 20.

Starting Netman

To begin using the AS/400 Limited Fault-tolerant agent, start the NETMAN process. To start NETMAN, log on as a user with *ALLOBJ authority on the AS/400 and enter the following commands:

CRTJOBD JOBД(QGPL/QMAESTRO) USER(QMAESTRO)
CHGOBJOWN OBJ(QGPL/QMAESTRO) OBJTYPE(*JOBD) NEWOWN(QMAESTRO)
CHGUSRPRF USRPRF(QMAESTRO) PASSWORD(password) JOBД(QGPL/QMAESTRO)

When you want to start the NETMAN job, sign on as the QMAESTRO User Profile and run:

STRNETMAN

Adding an AS/400 Limited Fault-tolerant Agent

To begin scheduling jobs on an AS/400 Limited Fault-tolerant agent, you must add it to the Tivoli Workload Scheduler network. To do this, add a CPU definition for the AS/400 using the Job Scheduling Console GUI program or the Job Scheduling Console command line.
Job Scheduling Console GUI Example

The following figure shows the Job Scheduling Console CPU Definition window for an AS/400 Limited Fault-tolerant agent named AS400-2. For information about using the console to add CPU definitions, refer to the Tivoli Job Scheduling Console User’s Guide.

![CPU Definition Window](image)

The fields in a CPU definition are as follows:

**CPU Name**
Specifies the Tivoli Workload Scheduler name of the AS/400 agent. The name must start with a letter and can contain alphanumeric, dash (-), and underscore (_) characters. With non-expanded databases, names can contain up to eight characters. With expanded databases, names can contain up to 16 characters.

**Node**
Specifies the node name or IP address of the AS/400.

**TCP Address**
Specifies the TCP port number of NETMAN on the AS/400 agent. The default is 31111. This must be the same as the value of the nm port option in the Local Options file. See “Local Options” on page 6 for more information about Local Options.

**Operating System**
Specifies the type of operating system. Select OTHER.

**Description**
A description of the agent. It can contain up to 40 alphanumeric characters.

**Domain**
Specifies the Tivoli Workload Scheduler domain of which the agent is a member. Click the Domains... button to display a list of available domains from which to select. The default is the master domain, usually named masterdm.

**Type**
Specifies the type of agent. Select Fault Tolerant Agent.
AUTO Link
Select AUTO Link to have the agent’s communication link opened automatically when its domain manager is started.

Ignore
Select this option to ignore this CPU definition. This is useful if you want to pre-define scheduling objects for an agent that is not installed.

Resolve Dependencies
Select this option to have the agent’s Production Control process operate in Resolve All Dependencies Mode. In this mode, the agent tracks dependencies for all jobs and schedules, including those running on other workstations.

If this option is not selected, the agent tracks dependencies for its own jobs and schedules only. This reduces CPU usage by limiting processing overhead.

To keep the agent’s Production Control file at the same level of detail as its domain manager select both Full Status and Resolve Dependencies options. See Full Status below.

Full Status
Select this option to have the link from the domain manager operate in Full Status mode. In this mode, the agent is kept updated about the status of jobs and schedules running on other workstations in the network.

If this option is not selected, the agent is only informed about the status of jobs and schedules on other workstations that affect its own jobs and schedules. This can improve operation by reducing network traffic.

To keep the agent’s Production Control file at the same level of detail as its domain manager select both Full Status and Resolve Dependencies options. See Resolve Dependencies above.

Server
Use this field to identify a Mailman server on the domain manager to handle communications with the agent. This can be a single letter or number (A-Z or 0-9). The IDs are unique to each domain manager, so you can use the same IDs for agents in different domains without conflict. If more than 36 server ids are required in a domain, consider dividing it into two or more domains.

The default is that messages to a fault-tolerant or standard agent are handled by a single Mailman process on the domain manager. Entering a server ID creates an additional Mailman process. The same server ID can be used for multiple agents. The use of servers reduces the time required to initialize agents, and improves the timeliness of messages. As a guide, additional servers should be defined to prevent a single Mailman from handling more than eight agents.

Command Line Example
Workstation definitions can also be entered at the Job Scheduling Console command line. The workstation definition shown in the previous example can be entered using the Job Scheduling Console command line program as follows:

cpuname as400-2
node 12.134.67.24
tcpaddr 31111
os other
domain masterdm
description "AS/400 fault-tolerant agent #2"
for maestro

Chapter 1. Installation and Configuration  5
Local Options

Local Options on the AS/400 Limited Fault-tolerant agent define the operating parameters for Tivoli Workload Scheduler processes. The Local Options are stored in a file named localopts. The file is installed automatically and contains Tivoli-supplied default values.

Setting Local Options

To modify the Local Options file, localopts, perform the following procedure:

1. Copy the localopts file using ftp to a UNIX or Windows NT workstation. For example:
   
   ```
   ftp 124.56.34.130
   User: root
   Password: xxxxx
   ftp> namefmt 1
   ftp> put /Maestro/localopts
   ftp> quit
   ```

   
   ```
   ftp 124.56.34.130
   User: root
   Password: xxxxx
   ftp> namefmt 1
   ftp> get localopts /Maestro/localopts
   ftp> quit
   ```

3. After copying the file back to the AS/400, use the following command to set the file ownership to QMAESTRO:
   
   ```
   CHGOWN OBJ('/maestro/localopts') NEWOWN(QMAESTRO)
   ```

Any changes do not take effect until Tivoli Workload Scheduler is stopped and restarted.

The syntax for the localopts file is described in the following table. Entries are not case-sensitive.

<table>
<thead>
<tr>
<th>Local Option Syntax</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td># comment</td>
<td></td>
</tr>
<tr>
<td>bm check file = seconds</td>
<td>120</td>
</tr>
<tr>
<td>bm check status = seconds</td>
<td>300</td>
</tr>
<tr>
<td>bm check until = seconds</td>
<td>300</td>
</tr>
<tr>
<td>bm look = seconds</td>
<td>30</td>
</tr>
<tr>
<td>bm read = seconds</td>
<td>15</td>
</tr>
<tr>
<td>bm stats = on</td>
<td>off</td>
</tr>
<tr>
<td>bm verbose = on</td>
<td>off</td>
</tr>
<tr>
<td>jm job table size = entries</td>
<td>160</td>
</tr>
</tbody>
</table>
### Local Option Syntax

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>jm look = seconds</code></td>
<td>300</td>
</tr>
<tr>
<td><code>jm nice = value</code></td>
<td>0</td>
</tr>
<tr>
<td>`jm no root = yes</td>
<td>no`</td>
</tr>
<tr>
<td><code>jm read = seconds</code></td>
<td>10</td>
</tr>
<tr>
<td>`merge stdlists = yes</td>
<td>no`</td>
</tr>
<tr>
<td><code>mm read = seconds</code></td>
<td>15</td>
</tr>
<tr>
<td><code>mm response = seconds</code></td>
<td>600</td>
</tr>
<tr>
<td><code>mm retry link = seconds</code></td>
<td>600</td>
</tr>
<tr>
<td>`mm sound off = yes</td>
<td>no`</td>
</tr>
<tr>
<td><code>mm unlink = seconds</code></td>
<td>960</td>
</tr>
<tr>
<td>`nm ipvalidate = none</td>
<td>full`</td>
</tr>
<tr>
<td>`nm mortal = yes</td>
<td>no`</td>
</tr>
<tr>
<td><code>nm port = tcpaddr</code></td>
<td>31111</td>
</tr>
<tr>
<td><code>nm read = seconds</code></td>
<td>60</td>
</tr>
<tr>
<td><code>nm retry = seconds</code></td>
<td>800</td>
</tr>
<tr>
<td><code>stdlist width = columns</code></td>
<td>80</td>
</tr>
<tr>
<td><code>syslog local = facility</code></td>
<td>-1</td>
</tr>
<tr>
<td><code>thiscpu = workstation</code></td>
<td>thiscpu</td>
</tr>
<tr>
<td><code>wr read = seconds</code></td>
<td>600</td>
</tr>
<tr>
<td><code>wr unlink = seconds</code></td>
<td>600</td>
</tr>
<tr>
<td><code>mozart directory = mozart_share</code></td>
<td>none</td>
</tr>
<tr>
<td><code>unison network directory = unison_share</code></td>
<td>none</td>
</tr>
<tr>
<td><code>parameters directory = parms_share</code></td>
<td>none</td>
</tr>
</tbody>
</table>

# comment

Treat everything from the pound sign to the end-of-line as a comment.

**bm check file**

Enter the minimum number of seconds Batchman will wait before re-checking for the existence of a file that is used as a dependency.

**bm check status**

Enter the number of seconds Batchman will wait between checking the status of an internetwork dependency.

**bm check until**

Enter the maximum number of seconds Batchman will wait before reporting the expiration of an Until time for job or job stream. Decreasing the value below the default setting may unduly load the system. If it is set below the value of Local Option `bm read`, the value of `bm read` is used in its place.

**bm look**

Enter the minimum number of seconds Batchman will wait before scanning and updating its Production Control file.

**bm read**

Enter the maximum number of seconds Batchman will wait for a message in its message file.
\textbf{bm stats}  
Enter \texttt{on} to have Batchman send its startup and shutdown statistics to its standard list file. Enter \texttt{off} to prevent Batchman statistics from being sent to its standard list file.

\textbf{bm verbose}  
Enter \texttt{on} to have Batchman send all job status messages to its standard list file. Enter \texttt{off} to prevent the extended set of job status messages from being sent to the standard list file.

\textbf{jm job table size}  
Enter the size, in number of entries, of the job table used by Jobman.

\textbf{jm look}  
Enter the minimum number of seconds Jobman will wait before looking for completed jobs, and performing general job management tasks.

\textbf{jm nice}  
UNIX only. Enter the nice value to be applied to jobs launched by Jobman.

\textbf{jm no root}  
UNIX only. Enter \texttt{yes} to prevent Jobman from launching root jobs. Enter \texttt{no} to allow Jobman to launch root jobs.

\textbf{jm read}  
Enter the maximum number of seconds Jobman will wait for a message in its message file.

\textbf{merge stdlists}  
Enter \texttt{yes} to have all Tivoli Workload Scheduler control processes, except Netman, write their console messages to a single standard list file. The file is given the name MAESTRO. Enter \texttt{no} to have each process write to its own standard list file.

\textbf{mm read}  
Enter the rate, in seconds, at which Mailman checks its mailbox for messages. If omitted, the default is 15 seconds. Defining a lower value will cause Tivoli Workload Scheduler to run faster at the expense of using more processor time.

\textbf{mm response}  
Enter the maximum number of seconds Mailman will wait for a response before reporting that another CPU is not responding. The response time should not be less than 90 seconds.

\textbf{mm retry link}  
Enter the maximum number of seconds Mailman will wait, after unlinking from a non-responding CPU, before it attempts to re-link to the CPU.

\textbf{mm sound off}  
Determines how Mailman will respond to a conman tellop \texttt{?} command. Enter \texttt{yes} to have Mailman display information about every task it is performing. Enter \texttt{no} to have Mailman send only its own status.

\textbf{mm unlink}  
Enter the maximum number of seconds Mailman will wait before unlinking from another CPU that is not responding. The wait time should not be less than the response time entered for the Local Option \texttt{nm response}. 
**nm ipvalidate**
Enter **full** to enable IP address validation and, if IP validation fails, the connection is not allowed. Enter **none** to allow connections when IP validation fails.

**nm mortal**
Enter **yes** to have netman quit when all of its child processes have stopped. Enter **no** to have Netman keep running when its child processes have stopped.

**nm port**
Enter the TCP port number that Netman responds to on this computer. This must match the TCP Address in the computer’s CPU definition.

**nm read**
Enter the maximum number of seconds Netman will wait for a connection request before checking its message queue for stop/start commands.

**nm retry**
Enter the maximum number of seconds Netman will wait before retrying a connection that has failed.

**stdlist width**
Defines the maximum width of Tivoli Workload Scheduler console messages. You can enter a column number in the range 1-255, and lines will be wrapped at that column or before, depending on the presence of imbedded carriage control characters. Enter a negative number, or zero, to ignore line width.

**syslog local**
For UNIX computers only. Enables or disables Tivoli Workload Scheduler system logging. Enter -1 to turn off system logging for Tivoli Workload Scheduler. Enter a number 0-7 to turn on system logging, and have TWS use the corresponding local facility (LOCAL0-LOCAL7) for its messages. Enter any other number to turn on system logging, and have Tivoli Workload Scheduler use the USER facility for its messages.

**thiscpu**
The Tivoli Workload Scheduler name of this CPU.

**wr read**
Enter the number of seconds Writer will wait for an incoming message before checking for a termination request from Netman.

**wr unlink**
Enter the number of seconds Writer will wait before exiting if no incoming messages are received. The lower limit is 120, and the default is 600.

**mozart directory**
Not used.

**unison network directory**
Not used.

**parameters directory**
Not used.

**Local Options File Example**
The following template file contains Tivoli’s default settings:

```
/maestro/config/localopts
```
During the installation process, a working copy of the Local Options file is installed as:

/maestro/localopts

You can customize the working copy to suit your requirements. The following is a sample Local Options file.

```bash
# Localopts file defines attributes of this CPU.
#-----------------------------------------------------
thiscpu = sys1
merge stdlists = yes
stdlibwidth = 80
sysloglocal = -1
#-----------------------------------------------------
# Attributes of this CPU for Batchman process:
bm check file = 120
bm check until = 300
bm look = 30
bm read = 15
bm stats = off
bm verbose = off
#-----------------------------------------------------
# Attributes of this CPU for jobman process:
jm job table size = 160
jm look = 300
jm nice = 0
jm no root = no
jm read = 10
#-----------------------------------------------------
# Attributes of this CPU for mailman process:
mm response = 600
mm retrylink = 600
mm sound off = no
mm unlink = 960
#-----------------------------------------------------
# Attributes of this CPU for netman process:
nm mortal = no
nm port = 31111
nm read = 60
nm retry = 800
#-----------------------------------------------------
# Attributes of this CPU for writer process:
wr read = 600
wr unlink = 720
#-----------------------------------------------------
# Optional attributes of this CPU for remote
# database files
# mozart directory = d:\tws\mozart
# parameters directory = d:\tws
# unison network directory = d:\tws..\unison\network
#
#-----------------------------------------------------
# End of localopts.
```

**Netman Configuration File**

The Netconf file defines the services provided by the Tivoli Workload Scheduler Netman process. The services are as follows:

- **2001** Start a writer process to handle incoming messages.
- **2002** Start TWS.
- **2003** Stop TWS.
2004 Find and return a standard list (stdlist) file to the requesting Console Management client.

The following is a listing of the file installed by Tivoli. Consult Tivoli Customer Support before making any changes in this file.
2001 client /qsys.lib/qtws.lib/writer.pgm
2002 son /qsys.lib/qtws.lib/mailman.pgm -parm 32000
2003 client /qsys.lib/qtws.lib/qstopsrv.pgm
2004 slinet /qsys.lib/qtws.lib/scribner.pgm

Uninstalling Tivoli Workload Scheduler

Note: Before uninstalling the product, stop all Tivoli Workload Scheduler processes. To do this, run the ENDTWS command and wait for all of the processes to stop.

To uninstall Tivoli Workload Scheduler, enter the following command:
DLTLICPGM LICPGM(5G97WKB)
Chapter 2. Operation

Defining AS/400 Jobs in Tivoli Workload Scheduler

The Limited Fault-tolerant Agent for AS/400 has two modes for executing jobs on the AS/400 system:

- Normal mode, which can execute any command on the AS/400, except SBMJOB, which should be avoided. When you run jobs in normal mode, all jobs on the system will acquire MONPROC as job name.
- Extended mode, which uses the “submit:” keyword in the command or script. You can execute any command in the AS/400, but the job will be executed through a SBMJOB command.

The only difference between the modes is the syntax of the command.

Note: If you want to take advantage of the SBMJOB command and its parameters, use the Extended mode which supports most of the SBMJOB parameters. Avoid the use of the SBMJOB command in Normal mode.

NOSTDLIST Keyword Usage

In normal use all spooled files generated by the jobs are enclosed in the stdlist. In instances where you do not want to have spooled files copied to the stdlist, sue the NOSTDLIST keyword. The following two examples shows how to use the NOSTDLIST keyword:

1. DOCOMMAND "CALL MSTRTEST/TESTONE NOSTDLIST:QPRINT"
   In this case the output of the program MSTRTEST/TESTONE produced in the QPRINT file will not be enclosed in the stdlist.

2. DOCOMMAND "DSPLIBL NOSTDLIST:QPRINT"
   In this case the output of the command as400 DSPLIBL produced in the QPRINT file will not be enclosed in the stdlist.

Job Status in Tivoli Workload Scheduler

Once Tivoli Workload Scheduler submits a job for execution, the job’s Tivoli Workload Scheduler state will change from READY to WAIT. A state of WAIT indicates that the job has been queued into the appropriate JOBQ, but that the job is not currently ACTIVE. Even if the job immediately transitions to ACTIVE, the Tivoli Workload Scheduler job state will remain in WAIT for one full “Check Interval.” The “Check Interval” is a configured parameter that defines the number of seconds between a job status check. The default value for “Check Interval” is 60 seconds. Once a job is ACTIVE, the Tivoli Workload Scheduler state of the job will change from WAIT to EXEC. At that point the Agent will wait for a normal or abnormal completion message from the job. If a normal completion message is received from the job, the Tivoli Workload Scheduler job state will change from EXEC to SUCC. An abnormal completion message will cause the state to change from EXEC to ABEND.

AS/400 Jobs Controlled by Tivoli Workload Scheduler: Normal Mode

A job definition is required for each AS/400 job you intend to schedule with Tivoli Workload Scheduler. A job definition includes the name of the job, the command
used to run the job, the name of the user with which the job is run, and job recovery options. The following figure shows a Job Definition window for a job named ASJOB2 that runs on an AS/400 agent named AS400-2.

For information about using the Job Scheduling Console program to define jobs, refer to the Tivoli Workload Scheduler User’s Guide.

The fields in the job definition windows are as follows:

**CPU** Specifies the Tivoli Workload Scheduler name of the AS/400 agent.

**Job Name** Specifies the Tivoli Workload Scheduler name of the AS/400 job. The name must start with a letter and can contain alphanumeric, dash, and underscore characters. Embedded spaces are not allowed. With non-expanded databases, the name can contain up to eight characters. With expanded databases, the name can contain up to 16 characters. This name is used to schedule the job in Tivoli Workload Scheduler.

**Logon** Specifies the name of the user for whom the job is run on the AS/400 agent. This must be a valid user on the AS/400 computer.

**Description** Specifies a description of the job. It can contain up to 40 alphanumeric characters.

**Script File** Enter the command that will be used to run the job on the AS/400 agent. For example:

CALL AP/APLJOB PARM(45)

**Note:** Avoid using the SBMJOB command in the Script File field and within a job specified in the Script File field. The SBMJOB command submits jobs independently that cannot be monitored by Tivoli Workload Scheduler.

**Command** Not used.
Interactive
Not used.

Recovery Options
For information about recovery options refer to the Tivoli Workload Scheduler User’s Guide.

Command Line Example
Jobs can also be defined using the Job Scheduling Console command line interface. The following example documents a Tivoli Workload Scheduler job named ASJOB2 that runs on AS/400 agent CPU AS400-2.

```
as400-2#gljob
streamlogon SALLY
scriptname "CALL GL/JGLJOB PARM('5' '20')"
description "GL job on AS/400 LFTA"
```

AS/400 Jobs Controlled by Tivoli Workload Scheduler: Extended Mode
With the Limited Fault-tolerant Agent for AS/400, Tivoli Workload Scheduler can launch jobs for execution on an AS/400 system. As with any other Tivoli Workload Scheduler job, execution will not begin until all specified dependencies have been resolved for the job. Through the agent, jobs are submitted for execution by the OS/400 SBMJOB command. As with running the SBMJOB command from the command line, unless otherwise specified, Tivoli Workload Scheduler launched jobs will run with the attributes of the User Profile used to start the AS400AGENT program.

AS/400 jobs are documented in Tivoli Workload Scheduler using the standard Tivoli Workload Scheduler job documentation window. The logon field can specify any valid and active AS/400 user profile. The parameters for an AS/400 job are specified in the command or script field of the job documentation window, using the following syntax. The braces and enclosed text would be replaced with the appropriate SBMJOB parameters for the job.

```
*submit: job(testjob) cmd(dsplib lib(qtws)) user(usera) jobd(qgpl/testjob)*
```

The following table lists the parameters associated with the OS/400 SBMJOB command. Not all of the parameters can be used with an Limited Fault-tolerant Agent AS/400 workstation. The table describes each parameter and whether the parameter is allowed by the AS/400 workstation.

<table>
<thead>
<tr>
<th>SBMJOB Parameter</th>
<th>Allowed</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMD()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>JOB()</td>
<td>Y</td>
<td>If not specified, the AS/400 job name will consist of the first 10 characters of the Tivoli Workload Scheduler job name.</td>
</tr>
<tr>
<td>JOBD()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>JOBQ()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>JOBPTY()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>SBMJOB Parameter</td>
<td>Allowed</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>OUTPTY()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>PRTDEV()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>OUTQ()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>USER()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>PRTTXT()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>RTGDTA()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>RQSDTA()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>SYSLIBL()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>CURLIB()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>INLLIBL()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>LOG()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>LOGCLPGM()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>INQMSGRPY()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>HOLD()</td>
<td>N</td>
<td>Tivoli Workload Scheduler submits all jobs on hold and then releases the jobs.</td>
</tr>
<tr>
<td>SCDDATE()</td>
<td>N</td>
<td>Tivoli Workload Scheduler performs the scheduling function.</td>
</tr>
<tr>
<td>SCDTIME()</td>
<td>N</td>
<td>Tivoli Workload Scheduler performs the scheduling function.</td>
</tr>
<tr>
<td>DATE()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>SWS()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>DSPSBMJOB()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>MSGQ()</td>
<td>N</td>
<td>All Tivoli Workload Scheduler jobs send completion messages to the QCOMPLETE message queue in the QTWS library.</td>
</tr>
<tr>
<td>SRTSEQ()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>SBMJOB Parameter</td>
<td>Allowed</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>LANGID()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>CNTRYID()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>CCSID()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>SBMFOR()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>JOBMMSGQMIX()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>JOBMMSGQFL()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
<tr>
<td>CPYENVVAR()</td>
<td>Y</td>
<td>See SBMJOB documentation for more information.</td>
</tr>
</tbody>
</table>

For example, to submit a job using the command `submit docommand` to a workstation named `as400-2`:
```
sbd as400-2#"submit: job(testjob) cmd(dsplib lib(qtws)) user(usera)
jobd(qGPL/testjobd);alias=job1;logon=userb"
```

Note that the result of this example will be that on the `as400-2` AS/400 machine the USERB user profile will be used to issue the command:
```
SBMJOB CMD(DSPLIB LIB(QTWS)) JOB(TEStJOB) USER(USERA) JOBD(QGPL/TESTJOB0D)
```

You must assure that USERB has the authorization necessary to issue this command on the AS/400.

**Scheduling AS/400 Jobs**

Jobs that run on AS/400 agents are scheduled in the same manner as other Tivoli Workload Scheduler jobs and can include time constraints and dependencies on other jobs, prompts, files, and resources. Use the Tivoli Job Scheduling Console to create a job stream (schedule) definition.

For information about using Job Scheduling Console to create schedules, refer to the *Tivoli Job Scheduling Console User’s Guide*.

**Command Line Example**

Schedules can also be entered using the Job Scheduling Console command line interface. The following example defines the `APSKED` schedule that runs on CPU `UX-1`:
```
SCHEDULE UX-1#APSKED
ON REQUEST
  FOLLOWS UX-1#GL.0
;
  AS400-1#APJOB1
  AS400-2#APJOB2
  FOLLOWS APJOB1
  APJOB3
  FOLLOWS APJOB2
END
```

---

*Chapter 2. Operation 17*
For information about using the Job Scheduling Console command line interface to create schedules, refer to the Tivoli Workload Scheduler Reference Guide.

## Controlling the Job Environment with JOBMANRC

To set up a generic environment for the execution of all jobs launched by Tivoli Workload Scheduler or pre-process their return codes, you can create a program named JOBMANRC in the QTWS library. The following is a template that can be used as a model for your JOBMANRC.

```plaintext
/*****************************************************/
/*                                                 */
/* JOBMANRC SAMPLE PROGRAM */                      */
/*                                                 */
/*****************************************************/

PGM PARM(&JCL_NAME &JCL_LEN)

/* Here we receive the name of the program to launch */
DCL VAR(&JCL_NAME) TYPE(*CHAR) LEN(128)
DCL VAR(&JCL_LEN) TYPE(*CHAR) LEN(3)
DCL VAR(&RC) TYPE(*CHAR) LEN(20) VALUE('')
DCL VAR(&CL_STATUS) TYPE(*CHAR) LEN(7) VALUE('')
DCL VAR(&USR_STATUS) TYPE(*CHAR) LEN(4) VALUE('')
DCL VAR(&MESSAGE) TYPE(*CHAR) LEN(255)
DCL VAR(&COMMAND) TYPE(*CHAR) LEN(255)
DCL VAR(&CMD_LEN) TYPE(*DEC) LEN(3)

/*****************************************************/
/* HERE WE CALL TO THE TWSEXEC UTILITY PROGRAM */
/*****************************************************/
CALL PGM(QTWS/TWSEXEC) PARM(&JCL_NAME &JCL_LEN &RC )
CHGVAR VAR(&CL_STATUS) VALUE(%SST(&RC 1 7))
CHGVAR VAR(&COMMAND) VALUE(%SST(&JCL_NAME 1 &CMD_LEN))

/*****************************************************/
/* HERE WE CALL TO THE TWSEXEC UTILITY PROGRAM */
/*****************************************************/
IF COND(&CL_STATUS *NE 'CPF0000') THEN(DO)

CHGVAR VAR(&MESSAGE) VALUE('QTWS/400: JOBMANRC ERROR -> ' +
*BCAT &CL_STATUS *BCAT ' executing the program: -> ' +
+BCAT &COMMAND)

/*****************************************************/
/* SEND A WARNING TO THE OPERATOR */
/*****************************************************/
SNDMSG MSG(&MESSAGE) TOUSR(*SYSOPR)

/*****************************************************/
/* SEND AN ERROR TO ALLOW THIS PROGRAM TO FAIL */
/*****************************************************/
SNDPGMMSG MSGID(CPF0006) MSGF(QCPFMSG) MSGTYPE(*ESCAPE)
ENDDO

CHGVAR VAR(&USR_STATUS) VALUE(%SST(&RC 12 4))

/*****************************************************/
/* VERIFY FOR A POSSIBLE ERROR IN EXECUTION */
/*****************************************************/
IF COND(&USR_STATUS *NE '0000') THEN(DO)

CHGVAR VAR(&MESSAGE) VALUE('QTWS/400: JOBMANRC PROGRAM USER ERROR -> ' +
+BCAT &USR_STATUS *BCAT ' executing the program: -> ' +
+BCAT &COMMAND)

/*****************************************************/
/* SEND A WARNING TO THE OPERATOR */
/*****************************************************/
SNDMSG MSG(&MESSAGE) TOUSR(*SYSOPR)

/*****************************************************/
/* SEND AN ERROR TO ALLOW THIS PROGRAM TO FAIL */
/*****************************************************/
```
If a JOBMANRC program exists in the QTWS library, Tivoli Workload Scheduler will execute it instead of executing the actual scheduled jobs. For each job, Tivoli Workload Scheduler will pass the scheduled job’s name to JOBMANRC in the variable &JCL_NAME. The scheduled job is executed by the Tivoli-supplied program TWSEXEC, which must be run in the manner shown in the template. TWSEXEC returns the scheduled job’s return code in the variable &RC, which can then be tested to determine the action to be taken.

JOBMANRC captures through TWSEXEC program, either the user program return code or the system message issued by it. If the return code is different from 0 or the system message is different from CPF0000 JOBMANRC sends an escape message to Tivoli Workload Scheduler in order to set the completion state of scheduled job to ABEND. If the return code is 0 and the system message by the user program is CPF0000, JOBMANRC does not send an escape message to Tivoli Workload Scheduler, so the completion state of the scheduled job will be set to SUCC.

Managing Tivoli Workload Scheduler on the AS/400 Limited Fault-tolerant Agent

Control of Tivoli Workload Scheduler on the AS/400 Limited Fault-tolerant agent is limited to starting and stopping the Tivoli Workload Scheduler processes. All other operations are performed on the Tivoli Workload Scheduler master domain manager.

Required Authority

To run the Tivoli Workload Scheduler commands on the AS/400, you must be a user with *ALLOBJ authority or have *USE authority for the QTWS library and *ALL authority for the objects in the library.

Tivoli Workload Scheduler Commands

The commands described here permit local control of Tivoli Workload Scheduler processes on the AS/400. For all other commands, use the Console Manager (conman) on the master domain manager. For more information about the Console Manager, refer to the Tivoli Job scheduling console User’s Guide.

STRNETMAN
    Run this command to start only the NETMAN process.

STARTTWS
    Run this command to start the remaining Tivoli Workload Scheduler processes after NETMAN has been started with STRNETMAN.

STOPTWS
    Run this command to stop all Tivoli Workload Scheduler processes except NETMAN.

ENDTWS
    Run this command to stop all Tivoli Workload Scheduler processes, including NETMAN, MAILMAN, and BATCHMAN.
Autostarting Tivoli Workload Scheduler
You can have Tivoli Workload Scheduler started when the AS/400 is rebooted. To do this, enter a sequence of commands like the following.
CRTJOBD JOBD(QGPL/QTWS) USER(QMAESTRO) RQSDTA(QTWS/STRNETMAN)
ADDAJE SBSD(QSYSWRK) JOB(STRNETMAN) JOBD(QGPL/QTWS)

Note: TCP/IP services must be loaded and running when NETMAN is started.

Managing Tivoli Workload Scheduler Production on the AS/400 Limited Fault-tolerant Agent
To manage Tivoli Workload Scheduler production on an AS/400 Limited Fault-tolerant agent, use the Console Manager running on the master domain manager. There is also a limited conman command line interface on the Limited Fault-tolerant agent AS/400. Most of the standard conman commands are available. The following restrictions apply:
• No shell program executions are allowed.
• No on-line help is available.
• There is no gconman program.
• There is no Remote Console access.
• Screen size for the AS/400 does not fit well for some display commands. No information is lost, you can view the information by scrolling the screen.
• If you type something wrong for a prompt that expects a yes or no answer, the job hangs.

For more information about managing production with the Console Manager, refer to the Tivoli Job Scheduling Console User’s Guide.

Displaying Tivoli Workload Scheduler Jobs on the AS/400 Limited Fault-tolerant Agent
To display Tivoli Workload Scheduler jobs on the AS/400, enter the following command:
WRKUSRJOB USER(QMAESTRO)

Note that all Tivoli Workload Scheduler jobs (NETMAN, JOBMAN, WRITER, etc) on AS/400 run in the QSYSWRK subsystem and can not be changed.

Tivoli Workload Scheduler Console Messages and Prompts
Tivoli Workload Scheduler production processes write status messages and console messages to standard list (stdlist) files. Included in these messages are the prompts used as job and schedule dependencies. Standard list files are located in the /maestro/stdlist directory.

Unsupported Tivoli Workload Scheduler Utilities
The QTWSTOOLS library contains the following unsupported utilities:

CAT.PGM
The CAT program can be used to display Tivoli Workload Scheduler standard list (stdlist) files. Usage:

CALL QTWSTOOLS/CAT PARM("filename")
where _filename_ is the name of the _stdlist_ file. For example, to display the Tivoli Workload Scheduler production _stdlist_ file for August 30, 1999, run the program as follows:

```
CALL QTWSTOOLS/CAT PARM('/maestro/stdlist/1999.08.30/QMAESTRO')
```

**CONMAN**

Conman is available on the AS/400 machine. It is qualified as unsupported because there are some limitations to its use. Conman is located in the QTWSTOOLS library. For more information about the limitations of the local conman on the Limited Fault-tolerant Agent for AS/400 refer to "Managing Tivoli Workload Scheduler on the AS/400 Limited Fault-tolerant Agent" on page 19.

**JAPJOB1.PGM**

The JAPJOB1 job can be used to demonstrate and test Tivoli Workload Scheduler. It runs for a specified period and then returns successful or abended status. Usage:

```
CALL QTWSTOOLS/JAPJOB1 PARM('status' 'secs')
```

where:

- _status_ Specifies the type of status you want the job to return. The values are:
  - 0: Return successful status.
  - <>0: Return abend status.

- _secs_ Specifies the number of seconds the job idles before ending.

For example, to return successful status after 30 seconds, run the job as follows:

```
CALL QTWSTOOLS/JAPJOB1 PARM('0' '30')
```

**MORESTDL**

The MORESTDL program is used to display the _stdlist_ for the current production date. Usage:

```
MORESTDL
```

This command is equivalent to the following:

```
CALL QTWSTOOLS/QDSPF PARM('/maestro/stdlist/YYYY.MM.DD/QMAESTRO')
```

For example, to display the _stdlist_ for the current production file:

```
MORESTDL
```

**QDSPF.PGM**

The QDSPF program can be used to display the Tivoli Workload Scheduler standard list file (_stdlist_) using the Source Entry Utility (SEU) of the AS/400. You can use this program to display any IFS streamed file. Usage:

```
CALL QTWSTOOLS/QDSPF PARM('filename')
```

For example, to display the _stdlist_ named _job_file_:

```
CALL QTWSTOOLS/QDSPF PARM('job_file')
```

**TERMINAL.PGM**

The TERMINAL program can be used to display _stdout_ and _stderr_ generated messages. Usage:

```
CALL QTWSTOOLS/TERMINAL
```
For example, to display the messages logged on **stdout** and **stderr** of the user’s session, run the program as follows:

```
CALL QTWSTOOLS/TERMINAL
```

**TOUCH.PGM**

The TOUCH program can be used to create the IFS files needed to demonstrate file dependencies. Usage:

```
CALL QTWSTOOLS/TOUCH PARM('filename')
```

where *filename* is the name of the created file. For example, to create `/tmp/testfile` with RW permissions, run the program as follows:

```
CALL QTWSTOOLS/TOUCH PARM('/tmp/testfile')
```
Chapter 3. Troubleshooting

This chapter contains troubleshooting information for this product.

Limitations and Workarounds

The following limitations are known:

- Deleting the Limited Fault-tolerant Agent for AS/400.
  
  Note that the DLTLICPGM command will fail if you have added any new objects, except the JOBMANRC program, in the QTWS library. If the DLTLICPGM command fails, proceed as follows:

1. Delete the /maestro directory as follows:
   WRKLINK OBJ('maestro')
   Use option 4.
2. Delete the /usr/unison directory as follows:
   WRKLINK OBJ('/usr/unison')
3. Delete the TWS commands in the QSYS directory as follows:
   DLTCMD QSYS/CUSTOMIZE
   DLTCMD QSYS/STRNETMAN
   DLTCMD QSYS/STARTTWS
   DLTCMD QSYS/STOPTWS
   DLTCMD QSYS/ENDTWS
4. Delete the QTWS library as follows:
   DLTLIB QTWS
5. Delete the QMAESTRO user as follows:
   DLTUSRPRF USRPRF (QMAESTRO) OWNBJOPT(+DLT)

- The IBM Tivoli Workload Scheduler Limited Fault-tolerant Agent for AS/400 Version 8.2 does not support the expanded database. Customize by setting the Expanded Version parameter to *YES. Use the Tivoli Workload Scheduler Limited Fault-tolerant Agent for AS/400 Version 8.2 only in an expanded database Tivoli Workload Scheduler network.

- Tivoli Workload Scheduler usually decides to set the state of an executed job depending on the invoked user program return code. If the user program return code is equal to 0 the executed jobs status is set to SUCC, otherwise it is set to ABEND. Note that the Tivoli Workload Scheduler Limited Fault-tolerant Agent for AS/400 Version 8.2 only receives the user program return code for C language based programs. For other kinds of program (for example, RPG based programs, and COBOL based programs) a better management can be performed using the Extended mode as described in “AS/400 Jobs Controlled by Tivoli Workload Scheduler: Extended Mode” on page 15.

- The Tivoli Workload Scheduler Limited Fault-tolerant Agent for AS/400 Version 8.2 has been shipped for compatibility reasons, it does not include any new feature implemented in the Tivoli Workload Scheduler Version 8.2, the implemented features are the same as the features implemented in the Tivoli Workload Scheduler Limited Fault-tolerant Agent for AS/400 Version 6.1.

- If the Tivoli Workload Scheduler Limited Fault-tolerant Agent for AS/400 Version 8.2 is installed on an AS/400 with primary language different from 2924 you could encounter some problems during the installation and running procedures. Use the following procedure as a workaround:
1. Install only the program part (not the language part) using:
   RSTLCPGM LICPGM(5G97WK8) DEV(SAVF) RSTOBJ(*PGM) LNG(2924)
   SAVF( *(1)LIBRARY_NAME/QFTASAVF )

   (*1) The library where the product save file was placed.

2. Install only the language in a different library (for example, the library QTWS_2924)
   RSTLCPGM LICPGM(5G97WK8) DEV(SAVF) LNG(*2924)
   SAVF( *(1)LIBRARY_NAME/QFTASAVF ) LNGLIB(QTWS_2924)

   (*1) The library where the product save file was placed.

3. Copy the following two objects from the QTWS_2924 (or the other langlib you chose) into the QTWS library:
   QFTALANG *PRDLOD
   QMAESTRO *MSGF

4. Set the authority for the above objects in the QTWS library as follows:
   QMAESTRO *ALL
   *PUBLIC *EXCLUDE

5. Run CUSTOMIZE

6. At this point you can check the installation using the command:
   DSPSFWRSC

   You will see:
   5G97WK8 *BASE 5001 Tivoli Workload Scheduler for OS400 8.2
   5G97WK8 *BASE 2924 Tivoli Workload Scheduler for OS400 8.2

   or using option 10 of the (GOLCPGM) command, you will see:
   5G97WK8 *NOPRIMARY Tivoli Workload Scheduler for OS400 8.2

7. Using an editor, for example, Edtf, open the (path) /maestro/NetConf, you will see its content as follows:
   # Netman configuration file for OS400 fault tolerant agent
   #
   # $Id: NetConf.400,v 1.1 1999/08/11 18:27:10 mjuarez Exp $
   #
   # each entry consists of 4 parts,
   # service request, service type, program, and options
   #
   # service type is client (can be multiple and get TCP circuit)
   #      son (only one at a time and not TCP circuit)
   #
   2001 client /QSYS.LIB/QTWS.LIB/WRITER.PGM
   2002 son /QSYS.LIB/QTWS.LIB/MAILMAN.PGM -parm 32000
   2003 client /QSYS.LIB/QTWS.lib/QSTOPSRV.PGM
   2004 client /QSYS.LIB/QTWS.lib/SCRIBNER.PGM

   Delete all commented lines starting with the ‘#’ character placed at the beginning of the file leaving only the line containing the service specification.

8. Perform all the operations described in "Starting Netman" on page 3.
Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.
Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

**Trademarks**

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both: AIX, BookManager, CICS, DB2, DB2 Universal Database, Hiperbatch, Hiperspace, IBM, the IBM logo, IMS, MVS, NetView, OS/390, RACF, SAA, Sysplex Timer, VTAM, and z/OS, Tivoli, the Tivoli logo 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.

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

A
authority 19
AUTO Link 5
autostarting TWS 20

B
bm check file 7
bm check status 7
bm check until 7
bm look 7
bm read 7
bm stats 8
bm verbose 8
books
feedback vi
online vi
ordering vi

C
CAT program 20
conventions
typeface ix
cpu definition
command line example 5
CPU definition
Composer 3
cpu name 4
Customer Support viii
CUSTOMIZE command 2, 23

directory names, notation ix
display TWS processes 20
Domain 4

e-mail contact viii
ENDTWS command 19
environment variables, notation ix

F
features 1
feedback about publications viii
Full Status 5

I
installation procedure 1

J
JAPJOB1 job 21
jm job table size 8
jm look 8
jm read 8
job definition 13
command line example 15
Job Name 14
JOBMANRC 18

L
Local Options 6
eample 9
Logon 14

M
manuals
feedback vi
online vi
ordering vi
merge stdlists 8
mm read 8
mm response 8
mm retry link 8
mm sound off 8
mm unlink 8

N
Netconf file 10
Netman configuration file 10
nm ipvalidate 9
nm mortal 9
nm port 9
nm read 9
nm retry 9
Node 4
notation
environment variables ix
path names ix
typeface ix

O
online publications
accessing vii
ordering publications vii

P
path names, notation ix
publications
feedback vi
online vi
ordering vi

R
Recovery Options 15
requirements for software 1
Resolve Dependencies 5
RSTLICPGM 1, 23

S
schedule definition
command line example 17
Script File 14
Server 5
software requirements 1
start NETMAN 3
STARTTWS command 19
stdlist width 9
STOPTWS command 19
STRNETMAN command 19

T
TCP Address 4
TERMINAL program 21
thiscpu 9
TOUCH program 22
TWS commands 19
TWS console messages 20
typeface conventions ix

U
uninstall TWS 11
unsupported utilities 3, 20
update 3

V
variables, notation for ix

W
wr read 9
wr unlink 9

© Copyright IBM Corp. 1991, 2004
Program Number: 5698-WSH

Printed in USA