IBM Workload Scheduler

Limited Fault-Tolerant Agent for IBM i

Version 9 Release 4

IBM Workload Scheduler

Limited Fault-Tolerant Agent for IBM i

Version 9 Release 4

Note

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

This edition applies to version 9, release 4, modification level 0 of IBM Workload Scheduler (program number 5698-WSH) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Figures

Tables

About this publication

Using IBM Workload Scheduler limited fault-tolerant agent for IBM i, you can schedule, launch, and control jobs on IBM i using the job scheduling tools of IBM Workload Scheduler.

What is new in this release

Learn what is new in this release.

For information about the new or changed functions in this release, see *IBM Workload Automation: Overview*, section *Summary of enhancements*.

For information about the APARs that this release addresses, see the IBM Workload Scheduler Release Notes at http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27048863 and the Dynamic Workload Console Release Notes at http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27048864.

New or changed content is marked with revision bars. For the PDF format, new or changed V9.4 content is marked in the left margin with a pipe (|) character and new or changed V9.4FP1 content is marked with an equal sign (=).

Who should read this publication

Learn the audience of this publication.

This publication is intended for system managers who will install and configure IBM Workload Scheduler limited fault-tolerant agents on IBM i, and system administrators who perform daily administration tasks.

Users of the publication should have some knowledge of:

- IBM Workload Scheduler
- IBM i operating system

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully.

With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

For full information, see the Accessibility Appendix in the *IBM Workload Scheduler User's Guide and Reference*.

Technical training

Cloud & Smarter Infrastructure provides technical training.

For Cloud & Smarter Infrastructure technical training information, see: http://www.ibm.com/software/tivoli/education

Support information

IBM provides several ways for you to obtain support when you encounter a problem.

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

- Searching knowledge bases: You can search across a large collection of known problems and workarounds, Technotes, and other information.
- Obtaining fixes: You can locate the latest fixes that are already available for your product.
- Contacting IBM Software Support: If you still cannot solve your problem, and you need to work with someone from IBM, you can use a variety of ways to contact IBM Software Support.

For more information about these three ways of resolving problems, see the appendix about support information in *IBM Workload Scheduler: Troubleshooting Guide*.

How to read syntax diagrams

Syntax diagrams help to show syntax in a graphical way.

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

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

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

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

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

The symbols have these meanings:

►►───

The statement begins here.

───►

The statement is continued on the next line.

►───

The statement is continued from a previous line.

```
───►◄
```

The statement ends here.

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

These are the conventions used in the diagrams:

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

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

- Optional items appear below the main path:

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

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

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

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

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

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

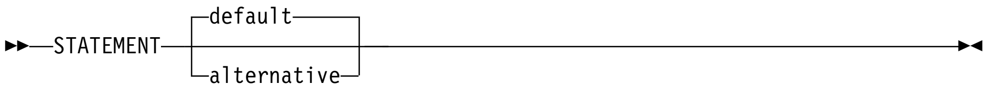
```
►►──STATEMENT──┬───────────────────┬───────────────────────────►◄
               ├─optional choice 1─┤
               └─optional choice 2─┘
```

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

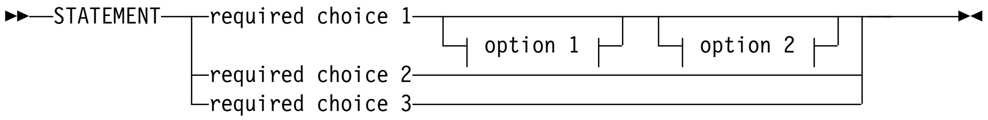
```
               ┌─,───────────────────┐
               ▼                     │
►►──STATEMENT────┬───────────────────┬┴────────────────────────►◄
                 ├─optional choice 1─┤
                 ├─optional choice 2─┤
                 └─optional choice 3─┘
```

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

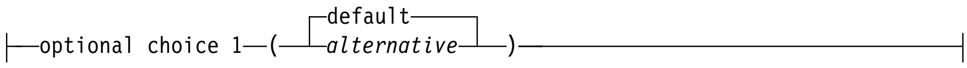
- Parameters that are above the main line are default parameters:



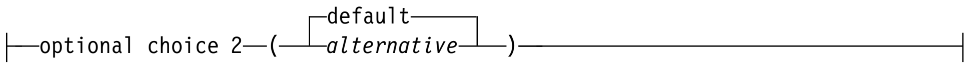
- Keywords appear in uppercase (for example, STATEMENT).
- Parentheses and commas must be entered as part of the command syntax, as shown.
- For complex commands, the item attributes might not fit on one horizontal line. If that line cannot be split, the attributes appear at the bottom of the syntax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



option 1



option 2



Chapter 1. Installing and configuring the agent

With the IBM Workload Scheduler limited fault-tolerant agent for IBM i (hereafter referred to as the IBM i agent), you can schedule, launch, and control jobs on an IBM i agent using the job scheduling tools of IBM Workload Scheduler.

The IBM i agent offers the following features:

- Allows the use of standard IBM Workload Scheduler job dependencies on IBM i jobs.
- Capability to schedule jobs on specific days and at specific times of day.
- Capability to prioritize jobs to have them executed in a prescribed order.
- Possibility to define interdependencies between IBM i jobs and jobs running on other systems, including UNIX, Windows NT, and SAP R/3.

IBM Workload Scheduler and the IBM i agent support Internet Protocol version 6 (IPv6), in addition to Internet Protocol version 4 (IPv4). To help you in the transition from an IPv4 environment to a complete IPv6 environment, IBM Workload Scheduler provides IP dual-stack support. This means that the product is designed to communicate using both IPv4 and IPv6 protocols simultaneously with other applications using IPv4 or IPv6.

Prerequisites

To install and use the IBM i agent, you must be running a supported version of IBM i. For up-to-date information about the supported versions of IBM i, see the IBM Workload Scheduler System Requirements Document at http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg24042843.

Before you install

To install the IBM i agent, you must be **QSECOFR** or have ***ALLOBJ** authority. In addition, the **QALWOBJRST** system value must be set to ***ALWPGMADP** or ***ALL**.

Installing the agent

About this task

This section describes the procedure to install the IBM i agent. Read the instructions carefully before starting the installation.

Only one instance of the IBM i agent can be installed on the same computer.

Note: Uninstall the product before attempting to install the software over a previous installation. For more information, see "Uninstalling IBM Workload Scheduler" on page 10.

Installing on IBM i systems where the primary language is set to 2924 using RSTLICPGM

If the system has the primary language is set to 2924, use the following procedure to install the IBM i agent using the **RSTLICPGM** command.

1. Insert the IBM i IBM Workload Scheduler Version 9.4 DVD into your UNIX or Windows system. On UNIX, mount the DVD using the appropriate operating system 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:

   ```
   cd twslftatmp
   ```
5. Copy and rename the IBM i agent product file into the temporary directory. For example:
   - On UNIX

     ```
     cp /MOUNT_DIR/As400/QTWS.V1R9M0.SAVF
     mv QTWS.V1R9M0.SAVF QFTASAVF
     ```

     where *MOUNT_DIR* is the directory under which the DVD was mounted.
   - On Windows

     ```
     copy X:\As400\QTWS.V1R9M0.SAVF
     ren mv QTWS.V1R9M0.SAVF QFTASAVF
     ```

     where *X* is the DVD letter identifier.
6. On IBM i, create a save file. For example:

   ```
   CRTSAVF QGPL/QFTASAVF
   ```

   where QGPL/QFTASAVF is the name of the save file.
7. On UNIX, use **ftp** to copy the save file to the IBM i save file. For example:

   ```
   ftp 124.56.34.130
   User: QSECOFR
   Password: xxxxx
   ftp> bin
   ftp> put QFTASAVF QGPL/QFTASAVF
   ftp> quit
   ```
8. On IBM i, run the **RSTLICPGM** command as follows:

   ```
   RSTLICPGM LICPGM(5G97WKB) DEV(*SAVF) SAVF(QGPL/QFTASAVF)
   ```
9. On IBM i, run the **CUSTOMIZE** command.
   a. For a new installation, use the following syntax:

      CUSTOMIZE TYPE(*NEW) THISCPU(*name***) MASTER(***name***) GROUP(***name***) COMPANY(***name***)**

      where:

      TYPE(*NEW)
      Required for new installations. Specifies that this is a new installation of the IBM i agent.

      THISCPU(*name***)**
      Required. Specifies the IBM Workload Scheduler name of the IBM i fault-tolerant agent. This name must be used later to formally define the workstation in IBM Workload Scheduler.

      MASTER(*name***)**
      Required. Specifies the IBM Workload Scheduler name of the master workstation.

      GROUP(*name***)**
      Required. Specifies the name of the product group in which to

install the IBM i fault-tolerant agent. 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**.

COMPANY(*name*)
Required. Specifies the name of your company.

For example:

```
CUSTOMIZE TYPE(*NEW) THISCPU('AS400-2') MASTER(CENTRAL) GROUP(PRODUCTION)
 COMPANY('AJAX FOODS')
```

Note: The CUSTOMIZE command with TYPE(*NEW) installs and customizes the IBM i IBM Workload Scheduler stream files. The files are installed in the new IBM i stream directories `/maestro` and `/usr/unison`. The stream directories are retrieved from template stream files stored in the QFTAFILES save file, that is an object of the QTWS library, the agent product library. You can run the CUSTOMIZE command with TYPE(*NEW) only once after the agent installation.

b. To update an existing customization, use the following syntax:

CUSTOMIZE TYPE(*UPDATE) THISCPU(*name* **) MASTER(***name***) GROUP(***name***)COMPANY(***name***)**

where:

TYPE(*UPDATE)
Specifies that this is an update of one or more of the four customization values entered the first time with TYPE(*NEW). This parameter is required for updating the customization of the IBM i agent.

THISCPU(*name*)
Optional. Specifies a new name for the IBM i agent workstation to be updated.

MASTER(*name*)
Optional. Specifies a new name for the master workstation to be updated.

GROUP(*name*)
Optional. Specifies a new name for the product group to be updated.

COMPANY(*name*)
Optional. Specifies a new name for the company name to be updated.

For example:

```
CUSTOMIZE TYPE(*UPDATE) THISCPU(TWA4) MASTER(TWMT) GROUP(TEST)
COMPANY(MY_COMPANY)
```

Note: You can run the CUSTOMIZE command with TYPE(*UPDATE) when necessary to update one or more of the values of the command parameters. These parameters are:

- Local CPU (**THISCPU**), stored in the `/maestro/localopts` IBM i stream file.
- Master CPU (**MASTER**) and Company name (**COMPANY**), both stored in the `/maestro/mozart/globalopts` IBM i stream file.
- Group name (**GROUP**), stored in the `/usr/unison/components` IBM i stream file.

10. To install the IBM 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, see "Unsupported utilities" on page 25.

Note: During the installation process, only the English messages catalogs are installed. All the output messages and reports of the IBM Workload Scheduler instance are in English.

Installing on IBM i systems where the primary language is not set to 2924 and on DBCS systems using RSTLICPGM

To install the IBM i agent using the **RSTLICPGM** command on systems where the primary language is not set to 2924 and on DBCS systems, perform the procedure described in "Installing on IBM i systems where the primary language is set to 2924 using RSTLICPGM" on page 1 by substituting Step 8 with the procedure described in IBM Workload Scheduler IBM i agent installed on System i® with primary language different from 2924 in Table 1 on page 27.

Note: During the installation process, only the English messages catalogs are installed. All the output messages and reports of the IBM Workload Scheduler instance are in English.

Starting Netman

About this task

To start using the IBM i agent, start the NETMAN process. Before you start NETMAN for the first time, you must sign on with a user that has ***ALLOBJ** special authority on the IBM i and enter the following commands:

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

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

```
STRNETMAN
```

Adding an IBM i agent

About this task

To begin scheduling jobs on an IBM i agent, you must add it to the IBM Workload Scheduler network. To do this, add a workstation definition for the System i using the Dynamic Workload Consoleor the **composer** command line.

For information about using the console to add workstation definitions, see the *Dynamic Workload Console User's Guide*.

For information about using the command line interface to create workstation definitions, see the *User's Guide and Reference*.

Local options

Local options on the IBM i agent define the operating parameters for IBM Workload Scheduler processes. The local options are stored in a file named **localopts**. The file is installed automatically and contains the supplied default values.

Setting local options

About this task

To modify the local options file, **localopts**, do what follows:

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
   ```

2. Edit the **localopts** file. For more information, see the *Administration Guide*.

   ```
   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 IBM i, use the following command to set the file ownership to **QMAESTRO**:

   ```
   CHGOWN OBJ('/maestro/localopts') NEWOWN(QMAESTRO)
   ```

Changes do not take effect until IBM Workload Scheduler is stopped and restarted.

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

Local option syntax	Default value
# *comment*	
bm check file = *seconds*	**120**
bm check status = *seconds*	**300**
bm check until = *seconds*	**300**
bm look = *seconds*	**30**
bm read = *seconds*	**15**
bm stats = on|off	**off**
bm verbose = on|off	**off**
jm job table size = *entries*	**160**
jm look = *seconds*	**300**
jm nice = *value*	**0**
jm no root = yes|no	**no**
jm read = *seconds*	**10**
merge stdlists = yes|no	**yes**
mm read = *seconds*	**15**

Local option syntax	Default value
mm response = *seconds*	**600**
mm retry link = *seconds*	**600**
mm sound off = yes|no	**no**
mm unlink = *seconds*	**960**
nm ipvalidate = none|full	**none**
nm mortal = yes|no	**no**
nm port = *tcpaddr*	**31111**
nm read = *seconds*	**60**
nm retry = *seconds*	**800**
stdlist width = *columns*	**80**
syslog local = *facility*	**-1**
thiscpu = *wkstation*	**thiscpu**
wr read = *seconds*	**600**
wr unlink = *seconds*	**600**
mozart directory = *mozart_share*	none
unison network directory = *unison_share*	none
parameters directory = *parms_share*	none

# 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 rechecking 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.

bm stats
Enter **on** to have Batchman send its startup and shutdown statistics to its standard list file. Enter **off** to prevent Batchman statistics from being sent to its standard list file.

bm verbose

Enter **on** to have Batchman send all job status messages to its standard list file. Enter **off** to prevent the extended set of job status messages from being sent to the standard list file.

jm job table size

Enter the size, in number of entries, of the job table used by Jobman.

jm look

Enter the minimum number of seconds Jobman will wait before looking for completed jobs, and performing general job management tasks.

jm nice

UNIX only. Enter the nice value to be applied to jobs launched by Jobman.

jm no root

For UNIX enter **yes** to prevent Jobman from launching root jobs. Enter **no** to allow Jobman to launch root jobs.

For IBM i enter **yes** to prevent Jobman from launching QSECOFR jobs. Enter **no** to allow Jobman to launch QSECOFR jobs.

jm read

Enter the maximum number of seconds Jobman will wait for a message in its message file.

merge stdlists

Enter **yes** to have all IBM Workload Scheduler control processes, except Netman, write their console messages to a single standard list file. The file is given the name MAESTRO. Enter **no** to have each process write to its own standard list file.

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 causes IBM Workload Scheduler to run faster at the expense of using more processor time.

mm response

Enter the maximum number of seconds Mailman will wait for a response before reporting that another workstation is not responding. The response time should not be less than 90 seconds.

mm retry link

Enter the maximum number of seconds Mailman will wait, after unlinking from a non-responding workstation, before it attempts to re-link to the workstation.

mm sound off

Determines how Mailman responds to a conman tellop ? command. Enter **yes** to have Mailman display information about every task it is performing. Enter **no** to have Mailman send only its own status.

mm unlink

Enter the maximum number of seconds Mailman will wait before unlinking from another workstation that is not responding. The wait time should not be less than the response time entered for the local option 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 workstation 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 IBM Workload Scheduler console messages. You can enter a column number in the range 1-255, and lines are 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

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

thiscpu

The IBM Workload Scheduler name of this workstation.

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 the default settings:

```
TWA_home/TWS/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.

```
# Localopts file defines attributes of this CPU.
#-----------------------------------------------------
thiscpu     =sys1
merge stdlists     =yes
stdlistwidth     =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 IBM 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 files installed by IBM Workload Scheduler. Consult 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 IBM Workload Scheduler

About this task

Note: Before uninstalling the product, you must end all IBM Workload Scheduler processes. To do this, stop and unlink the IBM i agent from its master, if necessary. Then, run the **ENDTWS** command and wait for all the processes to stop.

To uninstall IBM Workload Scheduler, enter the following command:

```
DLTLICPGM LICPGM(5G97WKB)
```

Chapter 2. Using the agent

This chapter describes how you define, schedule, and control IBM i jobs, as well as how you manage the IBM i agent.

Defining IBM i jobs

This section describes how you can define jobs to be run on an IBM i system.

Normal and extended modes

About this task

You define and schedule IBM i jobs at the IBM Workload Scheduler master domain manager level using the command-line interface and the generic syntax that is valid for any fault-tolerant agent processor. For example:

```
AS4CPU#AS4JOBNAME DOCOMMAND "command-string"
   STREAMLOGON "OS400 User Profile"
   DESCRIPTION "Test Description"
   RECOVERY STOP
```

where:

AS4CPU
: Is an IBM i processor defined in the IBM Workload Scheduler topology.

AS4JOBNAME
: Is the name of the job.

command-string
: Is a generic string consisting of a single IBM i user or system command.

OS400 User Profile
: Is an existing user profile.

The following sample shows a job definition for the TWAS processor:

```
TWAS#TEST_001 DOCOMMAND "WRKOUTQ OUTPUT(*PRINT)"
   STREAMLOGON "MINERMA"
   DESCRIPTION "Test N.1"
   RECOVERY STOP
```

When the TEST_001 job is submitted, it runs as a batch job on the IBM i processor in the same way as a user (logged into the system with user profile MINERMA) who entered the "WRKOUTQ OUTPUT(*PRINT)" command. This behavior is induced by the STREAMLOGON keyword with some limitations. For more information, see “Library list setting for IBM i jobs” on page 13.

The IBM i agent runs the command found in "command-string" in one of the following modes:

normal mode
: If the first token of "command-string" is not "SUBMIT:", the command is run in normal mode. See the job definition sample above where the WRKOUTQ command is run in normal mode. Normal mode can run and monitor any system and user command. In normal mode, for any submitted command, a dedicated IBM i agent job is started, with job name MONPROC and job user QMAESTRO. The submitted command is run and monitored until completion under the MONPROC job.

Do not use normal mode to run the SBMJOB system command. If "command-string" is SBMJOB, the command is run, but the monitoring is limited to the completion of the command and the job generated by SBMJOB is not monitored.

Use normal mode if you want to capture return codes from your jobs. The IBM i agent supplies a JOBMANRC user exit that allows you to manage return codes when normal mode is used. For details, see “Controlling the job environment with JOBMANRC” on page 21.

extended mode

If the first token of "command-string" is "SUBMIT:", the command is run in extended mode. To define a command to run in extended mode, set "command-string" to the SBMJOB command that must be submitted to run the same command on IBM i inside a new batch job. Use as many parameters as needed for SBMJOB, and then substitute SBMJOB with “SUBMIT:".

Not all the SBMJOB parameters are allowed in setting “command-string.” For more information, see “Controlling jobs in extended mode” on page 16.

Extended mode is not limited to monitoring a simple command but instead an IBM i job, meaning the SBMJOB system command. A dedicated MONPROC monitor job is started to run the SBMJOB command set in “command-string.” The monitor job waits until the completion of the related IBM i job, when MONPROC retrieves the system completion message and any related IBM i spool files. Depending on the completion message found into its dedicated QCOMPLETE message queue, the IBM i agent sets the ended job as *abended* or *successfully*. At this time, a report for the job is created in the stdlist directory of the IBM i agent, specifically the /maestro/stdlist stream directory, where the spool files are also included. Information about how to include or exclude the spool files is described in the following sections.

The following sample shows a job definition related to a job run in extended mode for the TWAS processor:

```
TWAS#TEST_002 DOCOMMAND "SUBMIT: CMD(WRKOUTQ OUTPUT(*PRINT))
JOB(MYJOB) USER(PEPPE)"
STREAMLOGON "MINERMA"
DESCRIPTION "Test N.2"
RECOVERY STOP
```

In this sample, the same system command of the previous sample, WRKOUTQ, is coded to run in extended mode. This means that the WRKOUTQ command is run inside a job with the job name MYJOB and the user name PEPPE. In the normal mode sample, the same command is run inside a job with job name MONPROC and user name QMAESTRO. In the SBMJOB command, if you omit the JOB or the USER parameter or both, the IBM i job inside of which the WRKOUTQ command is contained, runs with the job name TEST_002 and the job user MINERMA by default.

In both modes, the IBM i agent generates a job report. In extended mode only, the job report also includes an IBM i job log, if one was generated for the job.

Library list setting for IBM i jobs

About this task

The STREAMLOGON keyword does not have the same effect as an IBM i user profile that signs on to the system and issues a command. This is because the initial library list of the submitted job is, by default, that of the QMAESTRO user profile regardless of the STREAMLOGON value. To bypass the limitation of STREAMLOGON, use one of the following options:

Change the QMAESTRO job description to include all necessary libraries

You can use this option to avoid library list problems in either normal or extended mode. In this option, you stop the IBM i agent and change the QMAESTRO job description to include all necessary libraries in the IBM i library list.

Sign on with the QSECOFR user and end all the IBM i agent jobs. Then, change the initial library list in the QMAESTRO job description. *JOBD must be QMAESTRO in the QGPL library.

For example, if you want to add only the MINERMA library to the QMAESTRO initial library list, issue the following command

```
CHGJOBD JOBD(QGPL/QMAESTRO) INLLIBL(QGPL MINERMA)
```

After all of the necessary IBM i libraries are completed, you can restart the IBM i agent jobs.

In extended mode, use the SBMJOB parameters to access any necessary library

You can use this option to avoid library list problems only if you are in extended mode.

Use extended mode and explicitly set any necessary SBMJOB library-related keyword and the USER keyword, do not use default values when it is possible. Default values might not be sufficient to set the initial library list for a job run successfully under the IBM i agent. For example, consider the following job definition:

```
TWAS#TEST_003 DOCOMMAND "SUBMIT: CMD(CALL PGM(REPORTCL)) JOB(MYREPORT)
JOBD(*USRPRF) JOBQ(QSYS/QSYSNOMAX) CURLIB(*USRPRF) INLLIBL(*JOBD)"
STREAMLOGON "PEPPE"
DESCRIPTION "Test N.3"
RECOVERY STOP
```

Before you put the above "command-string" in the schedule of the IBM i agent for execution, verify which libraries are involved in the CALL of the REPORTCL program. Then, if these libraries are not included in the initial library list of the QMAESTRO job description, set the parameters USER, JOBD, CURLIB, and INLLIBL to values that make all the necessary libraries visible at job execution.

In this sample, assume the following:

- The REPORTCL program is inside the MINERMA library
- When started, the REPORTCL program makes internal calls to other programs inside two libraries: PEPPE and MINERMA
- PEPPE is the current library of the PEPPE user profile that does not have MINERMA in its job description initial library list
- MINERMA is the current library of the MINERMA user profile that does not have PEPPE in its job description initial library list

Considering the verified data above, the sample “command-string” must be modified so that both the PEPPE and MINERMA libraries are visible at job execution. You can do this in one of the following ways, depending on your specific needs and the execution environment in which the job runs:

- Modify INLLIBL(*JOBD) to INLLIBL(PEPPE MINERMA)
- Add the MINERMA library to the initial library list of the PEPPE job description
- Add the USER(MINERMA) parameter and add the PEPPE library to the initial library list of the MINERMA job description. Use this option only if the job must be executed under the MINERMA user profile.

NOSTDLIST keyword usage

About this task

All spooled files generated by the jobs are enclosed in the stdlist. There is a file type for any type of spool file on IBM i. For example the **QPRINT** file that collects any text message generated inside the programs of the job, like the text produced by the `printf` instruction inside any ILE-C program. Another example is the **QPRTLIBL** file, that holds the output produced by the **DSPLIBL IBM i** command. If you do not want to have one or more of these spooled files copied to the stdlist, use the **NOSTDLIST** keyword. The following examples show how to use the **NOSTDLIST** keyword:

- `DOCOMMAND "CALL MSTRTEST/TESTONE NOSTDLIST:QPRINT"`

  In this case, the output of the program MSTRTEST/TESTONE produced in the QPRINT file is not included in the stdlist.
- `DOCOMMAND "DSPLIBL NOSTDLIST:QPRTLIBL"`

  In this case, the output of the DSPLIBL command produced in the QPRTLIBL file is not included in the stdlist.
- `DOCOMMAND "SUBMIT: CMD(WRKACTJOB) NOSTDLIST: QPDSPAJB, QPJOBLOG"`

  In this case, the output of the WRKACTJOB command produced in the QPDSPAJB file is not included in the stdlist. Also, the joblog produced by the IBM i in the QPJOBLOG file is not included in the stdlist.

A list of items can follow the **NOSTDLIST** keyword. These items must be separated by commas.

LFTALOGS IBM i environment variable

About this task

The **NOSTDLIST** keyword option setting that works at job level might not be appropriate if you have a workload where the jobs produce many different spool file types that you need to exclude from the agent report when it consists of large spool files or includes security data. In these cases, you can use an option that works at IBM i agent level for any job to be submitted.

This option is based on the use of the IBM i system variable **LFTALOGS**. It forces the IBM i agent to ignore any spool file or, at least, include one or more of them. On the IBM i agent, you must create a new system level environment variable named LFTALOGS and set it to the value NONE to ignore any spool file. Or, you can set it to a list of the spool file types that are to be included. The list must begin with the “SPOOLS:” token as described below.

To force the IBM i agent to ignore any spool file, you must create the LFTALOGS variable as follows.

```
ADDENVVAR ENVVAR(LFTALOGS) VALUE(NONE) LEVEL(*SYS)
```

In this case, any IBM Workload Scheduler joblog report for the IBM i agent is reduced to the Header Trailer report.

To allow the IBM i agent to include only the QPRINT and the QPJOBLOG spool file types, that is, any spool files produced by “printf” instructions inside any ILE-C program and any produced joblog, you must create the LFTALOGS as follows.

```
ADDENVVAR ENVVAR(LFTALOGS) VALUE('SPOOLS: QPRINT QPJOBLOG') LEVEL(*SYS)
```

For any VALUE parameter set to an incorrect string, the IBM i agent will ignore the LFTALOGS environment variable option. When required, the LFTALOGS value can be changed with the CHGENVVAR system command. Both creation and change of the LFTALOGS environment variable can be done with the IBM iagent active, but with no workload activity.

Job status

When IBM Workload Scheduler submits a job to run, the job's IBM Workload Scheduler state changes 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 becomes ACTIVE, the IBM Workload Scheduler job state remains in WAIT for one full “Check Interval.” The “Check Interval” is a configured parameter that defines the number of seconds between job status checks. The default value for “Check Interval” is 60 seconds. Once a job is ACTIVE, the IBM Workload Scheduler state of the job changes from WAIT to EXEC. The IBM i agent then waits for a normal or abnormal completion message from the job. If a normal completion message is received, the job state changes from EXEC to SUCC; if an abnormal completion message is received, the state changes from EXEC to ABEND.

Exit status

The IBM i programming model was originally based on an early object orientation model in which programs communicated through message passing, rather than using return codes. The introduction of the Integrated Language Programming (ILE) model lead to the definitions of common areas to exchange data as return code: the user and the program return code. For information about the user return code, see “Controlling the job environment with JOBMANRC” on page 21.

When the IBM i agent verifies that a submitted command or job is completed, it assigns an exit status to the job based on the job status of the completed job. For more information, see “Job status.”

The exit status is set depending on the completion message of the command or job. For a normal completion, the exit status is set to zero. For an abnormal completion, it is set to the value of the severity of the message related to the exception that caused the abnormal end of the job. The IBM i agent can also set the exit status to the value of the user return code when it is returned by the submitted command in the normal mode. The exit status value is visible in the trailer of the IBM i agent report for the job and sent back to the IBM Workload Scheduler master as return code.

Defining IBM i jobs using the Dynamic Workload Console

In addition to using the command-line interface, you can also define an IBM i job by using the Dynamic Workload Console.

For information about using the console to add workstation definitions, see the *Dynamic Workload Console User's Guide*.

Controlling jobs in extended mode

About this task

With the IBM i agent, IBM Workload Scheduler can launch jobs to run on an IBM i system. As for any other IBM Workload Scheduler job, it does not begin to run until all specified dependencies have been resolved for the job. Using the IBM i agent, jobs are submitted to run in extended mode using the IBM i **SBMJOB** system command. In the same way as when running the **SBMJOB** command from the command line, unless otherwise specified, IBM Workload Scheduler launches jobs with the attributes of the user profile that was used to start the IBM i agent.

For more information about how to define the IBM i jobs, see the standard job documentation. The logon field can specify any valid and active IBM i user profile. The parameters for an IBM i job are specified in the command or script field of the IBM Workload Scheduler standard job, using the following syntax. The braces and enclosed text must be replaced with the appropriate SBMJOB parameters for the job.

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

The following table lists the parameters associated with the IBM i **SBMJOB** command. Not all of the parameters can be used on an IBM i system. The table describes each parameter and whether the parameter is allowed on an IBM i system.

SBMJOB parameter	Allowed	Description
CMD()	Y	See SBMJOB documentation.
JOB()	Y	If not specified, the IBM i job name consists of the first 10 characters of the IBM Workload Scheduler job name.
JOBD()	Y	See SBMJOB documentation.
JOBQ()	Y	See SBMJOB documentation.
JOBPTY()	Y	See SBMJOB documentation.
OUTPTY()	Y	See SBMJOB documentation.
PRTDEV()	Y	See SBMJOB documentation.
OUTQ()	Y	See SBMJOB documentation.
USER()	Y	See SBMJOB documentation.
PRTTXT()	Y	See SBMJOB documentation.
RTGDTA()	Y	See SBMJOB documentation.
RQSDTA()	Y	See SBMJOB documentation.
SYSLIBL()	Y	See SBMJOB documentation.
CURLIB()	Y	See SBMJOB documentation.
INLLIBL()	Y	See SBMJOB documentation.

SBMJOB parameter	Allowed	Description
LOG()	Y	See SBMJOB documentation. If not specified, by default this parameter is set to `LOG(4 10 *SECLVL)` by the IBM i agent.
LOGCLPGM()	Y	See SBMJOB documentation.
INQMSGRPY()	Y	See SBMJOB documentation.
HOLD()	N	IBM Workload Scheduler submits all jobs on hold and then releases the jobs.
SCDDATE()	N	IBM Workload Scheduler performs the scheduling function.
SCDTIME()	N	IBM Workload Scheduler performs the scheduling function.
DATE()	Y	See SBMJOB documentation.
SWS()	Y	See SBMJOB documentation.
DSPSBMJOB()	Y	See SBMJOB documentation.
MSGQ()	N	All IBM Workload Scheduler jobs send completion messages to the QCOMPLETE message queue in the QTWS library.
SRTSEQ()	Y	See SBMJOB documentation.
LANGID()	Y	See SBMJOB documentation.
CNTRYID()	Y	See SBMJOB documentation.
CCSID()	Y	See SBMJOB documentation.
SBMFOR()	Y	See SBMJOB documentation.
JOBMSGQMX()	Y	See SBMJOB documentation.
JOBMSGQFL()	Y	See SBMJOB documentation.
CPYENVVAR()	Y	See SBMJOB documentation.
ACGCDE()	Y	See SBMJOB documentation.
ALWMLTTHD()	Y	See SBMJOB documentation.
INLASPGRP()	Y	See SBMJOB documentation.
LOGOUTPUT()	Y	See SBMJOB documentation.
SPLFACN()	Y	See SBMJOB documentation.

For example, to submit a job using the conman **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
```

The result of this example is that on the **as400-2** IBM i machine the USERB user profile is used to issue the command:

```
SBMJOB CMD(DSPLIB LIB(QTWS)) JOB(TESTJOB) USER(USERA) JOBD(QGPL/TESTJOBD)
```

Ensure that USERB has the authorization necessary to issue this command on the IBM i.

Scheduling IBM i jobs

About this task

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

For information about using the Dynamic Workload Console to create job streams, see *Dynamic Workload Console User's Guide*.

File dependencies definition for database and stream files

About this task

The file dependencies mechanism on IBM i agents is the same as for all other IBM Workload Scheduler agents, with the exception of the **`opens`** keyword in association with IBM i files.

On IBM i agents, you can define file dependences based on two kind of physical files:

- Native database files
- Stream files (UNIX-like files)

You can use the **`opens`** keyword only with the **`-e %p`** qualifier for both file types. The rules to be used for the file name set as **`opens`** argument varies depending on the file type:

Native database files

Use the following syntax for the path name:

```
QSYS.LIB/library_name.LIB/file_name.FILE
```

where:

QSYS Is the file system that supports the i Series server library structure

library_name
Is the name of the library containing the file.

file_name
Is the name of the database file that is being checked for existence. The file name must be in capital letters.

Stream files (UNIX-like files)

The file name is case-sensitive.

The following example shows the definition of the SKEDSAMPLE schedule to be run on the `TWI5` workstation. This schedule includes two jobs, `OS4JOB1` and `OS4JOB2`, each one with file dependency.

`OS4JOB1` is submitted on `TWI5` if the `/usr/unison/components` stream file exists, and `OS4JOB2` is submitted on `TWI5` if the `QCLSRC` database file exists in the QGPL library:

```
SCHEDULE TWI5#SKEDSAMPLE
ON RUNCYCLE RULE1 "FREQ=DAILY;"
:
TWA41OS4JOB1
```

```
OPENS TWI5#"/usr/unison/components"(-e %p)
TWA41OS4JOB2
OPENS TWI5#"/QSYS.LIB/QGPL.LIB/QCLSRC.FILE"(-e %p)
END
```

For more information about file dependencies, see the sections about defining dependencies and the **opens** keyword in *IBM Workload Scheduler: User's Guide and Reference*.

The FILEWATCH function for the IBM i Agent

You can use the **filewatch** utility to check for file system changes on files and libraries, for example, when you want to make sure that a file exists before running an IBM i job that processes that file. By defining a job that runs the **filewatch** utility, you can implement file dependency, that is, a relationship between a file and an operation in which specific activity on the file determines the starting of the operation.

Using this utility, you can check for the following IBM i objects:

- Native libraries
- Native logical and physical files
- Native file members
- Stream files (UNIX-like files)

Syntax

FILEWATCH: COND(*condition***) FILE(***file_name***)**
DEADLINE(*deadline***) [INTERVAL(***interval***)]**
[RETCODE(*return_code***)]**

The parameters are not positional and the keywords are not case sensitive; for example, `COND`, `cond`, and `Cond` are equivalent. An open parenthesis must immediately follow each keyword without spaces, and it is not allowed the use of quotes to delimit the parameter values.

Parameters

COND

The condition to be checked. This is an required parameter. Valid values are:

wcr Waits until the object exists. If the object already exists, **FILEWATCH** exits immediately.

wmr Waits until the object size or modification time changes. If the object does not exists, **FILEWATCH** exits immediately.

wmc Checks that the object size or modification time stopped changing, meaning that **FILEWATCH** waits for three search intervals without any change. If the object does not exists, **FILEWATCH** exits immediately.

wmrc Waits until the object size or modification time changes and stops changing, meaning that, after the first change, **FILEWATCH** waits for three search intervals without any further change. If the object does not exists, **FILEWATCH** exits immediately.

wdl Stops running when the object is deleted. If the object does not exists, **FILEWATCH** exits immediately.

FILE The IBM i object to be processed. This is an required parameter. Use the following syntax depending on the type of object:

Native IBM i objects

- Libraries: *QSYS*.LIB/*library_name*.LIB
- Files: *QSYS*.LIB/*library_name*.LIB/*file_name*.FILE
- File members: *QSYS*.LIB/*library_name*.LIB/*file_name*.FILE/*member_name*.MBR

where:

- *QSYS* is the file system that supports the IBM i server library structure.
- *library_name* is the name of an IBM i library.
- *file_name* is the name of an IBM i file.
- *member_name* is the name of an IBM i file member.

All the above syntax elements are not case sensitive.

Stream files (UNIX-like files)

Use the following syntax:

path_name/*file_name*

where:

- *path_name* is the complete path of a UNIX directory.
- *file_name* is the name of a UNIX file.

The above syntax elements are case sensitive.

DEADLINE

The deadline period, expressed in seconds. This is an required parameter. Specify an integer in the range 30 to 31536000.

INTERVAL

The object search interval, expressed in seconds. This is an optional parameter. Specify an integer in the range 5 to 3600 that is not greater than the value specified for the **DEADLINE** parameter. The default is 30.

RETCODE

The exit return code, if the condition is not successfully checked by the deadline. This is an optional parameter. Specify an integer in the range 11 to 256. The default is 20.

Command line example

Job streams can also be entered using the command-line interface. The following example defines the **APSKED** job stream that runs on workstation **UX-1**:

```
SCHEDULE UX-1#APSKED
     ON REQUEST
     FOLLOWS UX-1#GL.@
:
  AS400-1#APJOB1
  AS400-2#APJOB2
     FOLLOWS APJOB1
  APJOB3
     FOLLOWS APJOB2
END
```

For information about using the command-line interface to create job streams, see the *User's Guide and Reference*.

Controlling the job environment with JOBMANRC

About this task

The normal mode is the only mode that allows retrieval of the end-of job codes, using the definition of a control user exit, JOBMANRC, described in this section as JCL source sample. See below for the meaning of each end-of-job code.

To set up a generic environment to run all jobs launched by IBM Workload Scheduler or pre-process their return codes, you can create a program named **JOBMANRC** in the **QTWS** library. The following is a template you can use as a model for your **JOBMANRC**.

```
/*************************************************************/
/*                                                           */
/*            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(4098)
DCL      VAR(&JCL_LEN) TYPE(*CHAR) LEN(4)
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(&PGM_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(4 0)
/*****************************************************/
/*   HERE WE CALL TO THE TWSEXEC UTILITY PROGRAM     */
/*****************************************************/
/* Now we call the CL Program that will launch the OPC script */
CALL     PGM(QTWS/TWSEXEC) PARM(&JCL_NAME &JCL_LEN &RC )
MONMSG   MSGID(CPF0000)
CHGVAR   VAR(&CL_STATUS) VALUE(%SST(&RC 1 7))
CHGVAR   VAR(&CMD_LEN) VALUE(&JCL_LEN)
CHGVAR   VAR(&COMMAND) VALUE(%SST(&JCL_NAME 1 &CMD_LEN))
/**********************************************/
/* VERIFY THE MESSAGE CODE AFTER EXECUTION    */
/**********************************************/
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 THE USER RETURN CODE AFTER 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)
ENDDO
CHGVAR   VAR(&PGM_STATUS) VALUE(%SST(&RC 8 4))
/************************************************/
/* VERIFY A PROGRAM RETURN CODE AFTER EXECUTION */
/************************************************/
IF     COND(&PGM_STATUS *GT '0001') THEN(DO)
CHGVAR VAR(&MESSAGE) VALUE('QTWS/400: JOBMANRC PROGRAM ERROR ->' +
*BCAT &PGM_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
EXITNOERR:
ENDPGM
```

If a **JOBMANRC** program exists in the **QTWS** library, IBM Workload Scheduler executes it instead of executing the actual scheduled jobs. For each job, IBM Workload Scheduler passes the scheduled job's name to **JOBMANRC** in the variable **&JCL_NAME**. The scheduled job is executed by the IBM Workload Scheduler 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.

TWSEXEC returns, for the scheduled job, a full return code that is the combination of the following end-of-job codes:

- End status code or <Status> (0 if successful)
- Program return code or <Prc> (0000 if successful)
- User return code or <Urc> (0000 if successful)

The full return code is a 15-character string that is the combination of status 'AAAnnnn', 'Prc', and 'Urc', where 'AAAnnnn' is 'CPF0000' when the end status code is zero (or an appropriate IBM i system message when the end status code is nonzero). 'AAAnnnn' is the System message associated to the end status code.

Note: On an IBM i system, the meaning of each end-of-job code is the following:

End status code
It indicates if the system issued a controlled cancellation of the job. Possible values are 1 (the subsystem or the job itself is canceled), 0 (the subsystem or the job itself is not canceled), blank (the job is not running).

Program return code
It specifies the completion code of the last program (such as a data file utility program, or an RPG or COBOL program, invoked by the job). If the job includes no program, the program return code is 0.

User return code
It specifies the user-defined return code set by ILE high-level language constructs. For example, the return code of a program written in C language. It represents the most recent return code set by any thread within the job.

JOBMANRC captures, using the **TWSEXEC** program, the full return code. This allows the end-of-job codes to be extracted and managed.

In the sample given, the first check is on the System message code: its only acceptable value is 'CPF0000'. The next check is on the User return code: its only acceptable value is 0. The last check is on the Program return code that can either be 0 or 1 (if any RPG, COBOL, or DFU programs are running). If any of the three return codes has a different value, **JOBMANRC** sends an escape message and the completion state of the scheduled job is set to ABEND. If all three return codes are acceptable, the escape message is not sent and the completion code of the job is set to SUCC.

Alternative method to retrieve the user return code with JOBMANRC

About this task

In some recent IBM i environments, the system API retrieving the *user return code* (urc) inside **TWSEXEC** does not retrieve the correct value for urc. It is therefore not recommended that you use any IBM i system APIs to retrieve the urc. To receive a value returned by a called program, it is better to provide, instead, a parameter to receive the value.

It is recommended that you use version 8.5.1 or later of user exit code, where a new urc retrieving "method" was implemented in the **TWSEXEC** code, with the following logic: a job environment variable is created, named **USERRC**, and set to the **INI** value before submitting the user command. When the command ends, **TWSEXEC** retrieves its urc with the system APIs, as usual, but it also verifies if the **USERRC** job environment variable was updated at user program level. If a value different from **INI** is found, this is considered as urc and the value retrieved through system APIs is ignored. This means that the user program modified the value of **USERRC**.

Modifying **USERRC** at user program level requires specific code in the application user code, where the urc is set to be returned. The following example shows how the user code passes back the urc via the IBM i agent reserved job environment variable **USERRC**:

```
----------------------------------------------------------
#include <stdio.h>
#include <stdlib.h> ---------------------------------(***)
#include <string.h> ---------------------------------(***)
void main(int argc, char *argv[])
{
  int      ind;
  int      iEndSts = 240;
  char     UserRetCode[10];
  system("DSPLIBL");
  ind = sprintf(UserRetCode, "%s", "USERRC=");
  ind += sprintf(UserRetCode+ind, "%d", iEndSts);
  putenv(UserRetCode);
  exit(iEndSts);
}
(***) >>> this is a required include <<<
----------------------------------------------------------
```

This example shows one way to set **USERRC** to the urc value. Usually, the urc is the result of complex processing, not just a simple system statement.

Managing the IBM i agent

Control of IBM Workload Scheduler on the IBM i agent is limited to starting and stopping the IBM Workload Scheduler processes. All other operations are performed on the IBM Workload Scheduler master domain manager.

Required authority

To run the IBM Workload Scheduler commands on the IBM i, 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.

IBM Workload Scheduler commands

The commands described here permit local control of IBM Workload Scheduler processes on the IBM i. For all other commands, use the Console Manager (**conman**) on the master domain manager. For more information about using **conman**, see the *User's Guide and Reference*.

STRNETMAN
Run this command to start only the NETMAN process.

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

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

ENDTWS
Run this command to stop all IBM Workload Scheduler processes, including NETMAN, MAILMAN, and BATCHMAN.

Autostarting IBM Workload Scheduler

About this task

You can have IBM Workload Scheduler start when the System i 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 production on the agent

About this task

To manage IBM Workload Scheduler production on an IBM i agent, use the Console Manager **conman** running on the master domain manager. There is also a limited **conman** command-line interface. Most of the standard **conman** commands are available. The following restrictions apply:

- No shell program executions are allowed.
- No online help is available.
- There is no **gconman** program.
- There is no Remote Console access.

- Screen size for the IBM i does not fit well for some display commands. No information is lost and you can scroll to view the information.
- If you type something wrong for a prompt that expects a yes or no answer, the job hangs.

Displaying jobs on the agent

About this task

To display IBM Workload Scheduler jobs on the IBM i, enter the following command:

```
WRKUSRJOB USER(QMAESTRO)
```

All IBM Workload Scheduler jobs, for example, NETMAN, JOBMAN, and WRITER, on IBM i run in the QSYSWRK subsystem and cannot be changed.

Messages and prompts

IBM 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 utilities

The QTWSTOOLS library contains the following unsupported utilities:

CAT.PGM

The CAT program can be used to display IBM 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 IBM 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 IBM i 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 IBM i agent, see “Managing the IBM i agent” on page 24

JAPJOB1.PGM

The JAPJOB1 job can be used to demonstrate and test IBM 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 IBM Workload Scheduler standard list file (**stdlist**) using the Source Entry Utility (SEU) of the IBM i. 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')
```

RMDIR.PGM

The RMDIR program can be used to recursively delete UNIX directories on IBM i. Usage:

CALL PGM(QTWSTOOLS/RMDIR) PARM('/maestro/*mydir*')

where *mydir* is the directory to be deleted.

The following job sample, to be submitted by the IBM i agent in extended mode, uses RMDIR to delete /maestro/stdlist or any of its subdirectories:

```
TWAS#RMDIR DOCOMMAND "submit: cmd(CALL PGM(QTWSTOOLS/RMDIR)
    PARM('/maestro/stdlist/2007.03.16')) USER(QMAESTRO)"
   STREAMLOGON "QMAESTRO"
   DESCRIPTION "Test RMDIR"
   RECOVERY STOP
```

Chapter 3. Troubleshooting

This chapter contains troubleshooting information for this product.

Limitations, problems, and workarounds

Table 1 lists the known problems and limitations, together with their solution or workaround.

Table 1. Limitations, problems, and workarounds

Limitation or problem	Workaround

The DLTLICPGM command, which deletes the IBM i agent, fails if you 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:
	`WRKLINK OBJ('maestro')`
	Use option 4.
	2. Delete the /usr/unison directory:
	`WRKLINK OBJ('/usr/unison')`
	3. Delete the IBM Workload Scheduler commands in the QSYS directory:
	`DLTCMD QSYS/CUSTOMIZE`
	`DLTCMD QSYS/STRNETMAN`
	`DLTCMD QSYS/STARTTWS`
	`DLTCMD QSYS/STOPTWS`
	`DLTCMD QSYS/ENDTWS`
	4. Delete the QTWS library:
	`DLTLIB QTWS`
	5. Delete the QMAESTRO user:
	`DLTUSRPRF USRPRF (QMAESTRO) OWNOBJOPT(*DLT)`

Problem with how IBM Workload Scheduler sets the state of run jobs.	IBM Workload Scheduler usually sets the state of a run job based on the invoked user program return code. If the user program return code is 0, the run job status is set to SUCC, otherwise it is set to ABEND. Note that the IBM Workload Scheduler IBM i agent receives the user program return code only for C language-based programs. For other types of program (for example, RPG-based and COBOL-based), use the extended mode, as described in "Controlling jobs in extended mode" on page 16.

Table 1. Limitations, problems, and workarounds (continued)

Limitation or problem	Workaround

If the IBM Workload Scheduler IBM i agent is installed with primary language different from 2924, you might encounter problems during the installation and running procedures.	When this happens, use the following procedure as a workaround:
	steps for DBCS systems or systems where primary language is not 294
	1. Install only the program part (not the language part) using:
	`RSTLICPGM LICPGM(5G97WKB) 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)
	`RSTLICPGM LICPGM(5G97WKB) DEV(*SAVF) RSTOBJ(*LNG)`
	`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:
	`QMAESTRO *ALL`
	`*PUBLIC *EXCLUDE`
	5. In the QMAESTRO user profile, set CCSID 37 and LANGID ENU.
	6. Run CUSTOMIZE.
	7. Check the installation either using the command DSPSFWRSC, you see:
	`5G97WKB *BASE 5001 IBM Workload Scheduler for IBM i`
	`5G97WKB *BASE 2924 IBM Workload Scheduler for IBM i`
	or using option 10 of the (`GOLICPGM`) command, you see:
	`5G97WKB *NOPRIMARY IBM Workload Scheduler for IBM i`
	8. Using an editor, for example, Edtf, open the (path) /maestro/NetConf to see its content:
	`# Netman configuration file for IBM i 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 at the beginning of the file leaving only the line containing the service specification.
	9. Perform all the operations described in “Starting Netman” on page 4.

Table 1. Limitations, problems, and workarounds (continued)

Limitation or problem	Workaround

When IBM i agent `*.msg` stream files are created, at installation time, their size is 1MB and there is no documented procedure or provided utility to increase the size.	To increase the size of `*.msg` stream files, add an environment variable, EVSIZE, in the IBM i system by issuing:
	`> ADDENVVAR ENVVAR(EVSIZE) VALUE(10000000) LEVEL(*SYS)`
	To do this, sign on to the IBM i system with a user that has *JOBCTL special authority, for example, the QSECOFR user. The IBM i agent interprets the EVSIZE variable which, in the previous sample, was set to 10MB.
	The workaround works only if no job is active on the IBM i agent and if the `*.msg` stream files are not already in the /maestro and /maestro/pobox directories. If the `*.msg` stream files to be resized already exist, they must be deleted before resizing.
	To check, change, or remove the EVSIZE variable, enter the following command as an QSECOFR user:
	`WRKENVVAR LEVEL(*SYS)`

A conman stop for the IBM i agent issued from the master domain manager also causes the writer to stop running on the IBM i agent and to lose the link. This does not affect linking at Jnextday/Jnextplan.	This is a known limitation. If `conman stop LFTA` is issued from the master domain manager, use `conman link LFTA` to reestablish the link.
Only one instance of the IBM i agent can be installed. The QMAESTRO is hardcoded.	None.
When running a job containing the following command: `CHGJOB CCSID (65535)` The Job log is not returned when asked from the master.	Modify `CCSID` back to 37 before ending the job and carry out the following steps: 1. Start Program 2. Execute `CHGJOB CCSID (65535)` 3. Program BODY 4. Execute `CHGJOB CCSID(37)` 5. End Program

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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 grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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 jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide 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 Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

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 is for planning purposes only. The information herein is subject to change before the products described become available.

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 actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

© (your company name) (year).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM® Corp. and Quantum in the U.S. and other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

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

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

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

ITIL is a Registered Trade Mark of AXELOS Limited.

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

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Index

Product Number: 5698-WSH

Printed in USA