IBM

IBM® Workload Scheduler

Scheduling Job Integrations with IBM® Workload Scheduler

Version 10.2.4

Note

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

This edition applies to version 10, release 2, modification level 4 of IBM® Workload Scheduler (program number 5698-T09) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

List of Figures

List of Tables

About this Publication

This guide provides information about how to set up and use job integrations to extend workload scheduling capabilities, access methods that run and control PeopleSoft, SAP, and z/OSjobs and the integration with SAP.

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 and the Dynamic Workload Console Release Notes. For information about the APARs addressed in a fix pack, refer to the readme file for the fix pack.

New or changed content is marked with revision bars.

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 ' [ SUBSYS ( { OPCA | subsystem name | MSTR } ) ] [ AVAIL ( { KEEP | RESET | NO | YES } ) ] [
DEVIATION ( { KEEP | amount | RESET } ) ] [ QUANTITY ( { KEEP | amount | RESET } ) ] [ CREATE ( { YES | NO } ) ] [ TRACE ( { 0 |
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:

```
STATEMENT [{default  | alternative}]
```

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

```
STATEMENT {required choice 1 [optional choice 1 ( {default | alternative} ) ] [optional choice 2 ( {default |
alternative} ) ] | required choice 2 | required choice 3}
```

Part I. Job integrations to extend workload scheduling capabilities

A wide variety of out-of-the-box adaptors or integrations are provided to integrate your business processes. The job integrations allow you to orchestrate Enterprise Resource Planning and Business Intelligence solutions (PeopleSoft, Oracle E-Business, Salesforce) and other business related systems. New applications are added to your organization all the time. By integrating them into your existing IBM Workload Scheduler environment you save time in getting skilled on new applications because you can administer them just like any of your existing jobs.

By extending the concept of jobs and workload scheduling to other applications you can continue to define jobs for your business process, add them to job streams, submit them to run according to schedule, and then monitor any exceptions all from a single entry point. The job integrations require an IBM Workload Scheduler dynamic agent, IBM Z Workload Scheduler Agent (z-centric), or both. For more information, see Supported agent workstations.

Note: Some of the old integrations previously provided with the product, are now out-of-the-box integrations available on Automation Hub. The related documentation has been removed from the product library and has been made available on Automation Hub.

In addition to these job integrations, you can find new integrations on Automation Hub that extend your automation processes.

The following sections provide an overview of creating job definitions and job streams, submitting them to run, monitoring them, and then analyzing the job log and job output. These procedures can be applied to any of the supported job integrations.

For information about the supported versions of the job integrations, generate a dynamic Data Integration report from the IBM® Software Product Compatibility Reports web site, and select the Supported Software tab.

Tip: Many of the IBM Workload Scheduler job integrations are illustrated in helpful, how-to demonstrations videos available on the Workload Automation YouTube channel.

Chapter 1. Defining a job

Define IBM Workload Scheduler jobs to run business tasks and processes defined in an external application.

Define your IBM Workload Scheduler job to run tasks or processes you have defined in external applications. Using the IBM Workload Scheduler job plug-in for your external application, you can define, schedule and run jobs to automate your business.

Distributed In distributed environment, define a job by using the Dynamic Workload Console connected to a distributed engine, by using the **composer** command line.

z/OS In a z/OS environment, define a job by using the Dynamic Workload Console connected to a z/OS engine.

How to define a job using the Dynamic Workload Console

For details about defining jobs from the Dynamic Workload Console, see the section about creating job definitions in *Dynamic Workload Console User's Guide*.

How to define a job using the composer command line

The composer command line supports the following syntax when defining a job:

$jobs
[*[folder/]workstation#*][*folder/]jobname*
{**scriptname** *filename* **streamlogon** *username* |
docommand "*command*" **streamlogon** *username* |
task *job_definition* }
[**description** "*description*"]
[**tasktype** *tasktype*]
[**interactive**]

[**succoutputcond** *Condition_Name* "*Condition_Value*"]
[**outputcond** *Condition_Name* "*Condition_Value*"]

[**recovery**
{**stop**
[**after** [*[folder/]workstation#*][*folder/]jobname*]
[**abendprompt** "*text*"]]
|**continue**
[**after** [*[folder/]workstation#*][*folder/]jobname*]
[**abendprompt** "*text*"]]
|**rerun [same_workstation]**
[[**repeatevery** *hhmm*] [**for** *number* **attempts**]]
[**after** [*[folder/]workstation#*][*folder/]jobname*]
|[**after** [*[folder/]workstation#*][*folder/]jobname*]
[**abendprompt** "*text*"]}

Use the **task** argument, specifying the XML syntax for the specific job plug-in. See the section for each job plug-in for the specific XML syntax.

For a detailed description of the XML syntax, see the section about job definition in *User's Guide and Reference*.

For some jobs a properties file can be generated and used to provide the values for some of the properties defined in the job definition.

The properties file is automatically generated either when you perform a "Test Connection" from the Dynamic Workload Console in the job definition panels, or when you submit the job to run the first time. Once the file has been created, you can customize it. This is especially useful when you need to schedule several jobs of the same type. You can specify the values in the properties file and avoid having to provide information such as credentials and other information, for each job. You can override the values in the properties files by defining different values at job definition time.

Chapter 2. Scheduling and submitting jobs and job streams

You schedule IBM Workload Scheduler jobs by defining them in job streams.

Distributed For distributed environments, use the Dynamic Workload Consoleor the conman command line.

After you define an IBM Workload Scheduler job, add it to a job stream with all the necessary scheduling arguments and submit it to run. After submission, when the job is running (**EXEC** status), you can kill the IBM Workload Scheduler job if necessary. For some job plug-ins, this action is converted into corresponding action in the plug-in application. Refer to the specific plug-in section for details about what effect the kill action has in the application.

z/OS For z/OS environments, use the Dynamic Workload Console or the ISPF application.

How to submit a job stream using the Dynamic Workload Console

To submit a job or job stream to run according to the schedule defined, see the section about submitting workload on request in production in *Dynamic Workload Console User's Guide*. For distributed environments only, see also the section about quick submit of jobs and job streams in *Dynamic Workload Console User's Guide*.

How to submit a job stream from the conman command line

To submit a job stream for processing, see the **submit sched** command. To submit a job to be launched, see the **submit job** command. For more information about these commands see the *IBM Workload Scheduler: User's Guide and Reference*.

How to submit your workload using the ISPF application

The workload is defined by creating one or more calendars, defining applications, creating a long-term plan, and creating a current plan. The current plan is a detailed plan, typically for one day, that lists the applications that run and the operations in each application. See the section about creating the plans for the first time in *Managing the Workload* for more information about creating plans.

Chapter 3. Monitoring IBM Workload Scheduler jobs

Monitor IBM Workload Scheduler jobs by using the Dynamic Workload Console, the command line, or the ISPF application.

Distributed You monitor distributed jobs by using the Dynamic Workload Console connected to a distributed engine, by using the **conman** command line.

z/OS You monitor z/OS jobs by using the Dynamic Workload Console connected to a z/OS engine or the ISPF application.

How to monitor jobs by using the Dynamic Workload Console

See the online help or the section about creating a task to monitor jobs in the *Dynamic Workload Console User's Guide*.

How to monitor jobs by using conman

See the section about managing objects in the plan - conman in *User's Guide and Reference*.

How to monitor jobs by using the ISPF application

See the section about monitoring the workload in *IBM® Z Workload Scheduler Managing the Workload*.

Chapter 4. Analyzing the job log

When a job runs IBM Workload Scheduler creates a job log that you can analyze to verify the job status.

About this task

Distributed For distributed jobs, you analyze the job log by using the Dynamic Workload Console or the **conman** command line.

z/OS For z/OS jobs, you analyze the job log by using the Dynamic Workload Console or the ISPF application.

While the job is running, you can track the status of the job and analyze the properties of the job. In particular, in the Extra Information section, if the job contains variables, you can verify the value passed to the variable from the remote system. Some job streams use the variable passing feature, for example, the value of a variable specified in job 1, contained in job stream A, is required by job 2 in order to run in the same job stream.

For more information about passing variables between jobs, see the related section in the IBM Workload Scheduler on-premises online product documentation in IBM Knowledge Center.

How to analyze the job log using the Dynamic Workload Console

Before you can access the job log for an individual job, you need to run a query and list the jobs for which you want to analyze the job log. See the online help or the section about creating a task to monitor jobs in *Dynamic Workload Console User's Guide*. From the list of jobs resulting from the query, you can either download the job log, or view the job log in the job properties view. Select the job for which you want to analyze the job log and click **More Actions > Download Job Log** or **More Actions > Properties** from the toolbar.

How to analyze the job log using conman

See the section about the **showjobs** command in *User's Guide and Reference*.

How to analyze the job log using the ISPF application

See the section about monitoring the workload in *Managing the Workload*.

Part II. Access methods

Access methods are used to extend the job scheduling functions of IBM Workload Scheduler to other systems and applications. They run on extended agents, dynamic agents, and IBM Z Workload Scheduler agents. They enable communication between external systems (SAP R/3) and IBM Workload Scheduler and launch jobs and return the status of jobs.

For information about the supported versions of the plug-ins and access methods, run the Data Integration report and select the **Supported Software** tab.

PeopleTools 8.61 and later require connecting using TLS. To continue working with an unsecure connection without enabling TLS, modify the `psagent` script by adding the **TM_ALLOW_NOTLS** option and setting it to `yes`.

Chapter 5. Installing and configuring the access methods

The access methods documented in this guide are packaged with IBM Workload Scheduler and are automatically installed with the product on dynamic and fault-tolerant agents.

Important: In order to be entitled to use the access methods and plug-ins, you must have purchased at least one of the following offerings: IBM Workload Scheduler, IBM Workload Scheduler for Applications, or IBM Z Workload Scheduler Agent. See the 10.2.4 Quick Start Guide available from IBM Fix Central. For information about the supported versions of the plug-ins and access methods, open the Data Integration report and select the **Supported Software** tab.

For details about installing an IBM Workload Scheduler dynamic or fault-tolerant agent, see IBM® Workload Scheduler Planning and Installation.

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

Setting options for the access methods

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

```
option=value
```

All access methods use two types of options files: a global options file and one or more local options files. The names of the local options files are generically referred to as `XA_Unique_ID`_accessmethod_.opts on extended agents and `DYNAMIC_AGENT_FILE`_accessmethod_.opts on dynamic agents. The file names specified for the local options files for both types of agents must respect the following rules:

- Both `XA_Unique_ID` and `DYNAMIC_AGENT_FILE` in the file name must be uppercase alphanumeric characters. See specific requirements about `XA_Unique_ID` in XA_Unique_ID on page 21.
- Double-byte character set (DBCS), single-byte character set (SBCS), and bidirectional text are not supported. For information about acceptable values for the extended agent workstation name, See Table 1: How to complete the extended agents definition on page 23.

Dynamic agents and IBM Z Workload Scheduler agents

Global options file

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

For PeopleSoft

psagent.opts

For SAP

r3batch.opts

Local options file

One or more configuration files that are specific to each access method. The name of this file is *optionsfile_accessmethod*.opts and they are saved to the path *TWA_DATA_DIR*/methods.

In a distributed environment

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

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

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

If the SAP access method is installed for AGENT1 workstation, with unique identifier, S4HANAR3BW, but you have two external SAP systems on which to schedule jobs, then in the *TWA_DATA_DIR*/methods directory, you create the following options files:

- SAP1_S4HANAR3BW_r3batch.opts
- SAP2_S4HANAR3BW_r3batch.opts

Each file contains the options specific to each external SAP system, for example, the connection information.

For pools and dynamic pools containing *n* agents, you must create an options file for the dynamic pool and copy it in the *TWA_DATA_DIR*/methods of each agent of the pool so that all members of the pool have a local options file with the same name. Then you must create another options file for the specific agent in the same directory. For example, if the SAP access

method is installed for **AGENT1** and **AGENT2** which belong to the dynamic pool **DYN_POOL**, create in the *TWA_DATA_DIR*/methods directory of each agent the following options files:

AGENT1

- FILEOPTS_AGENT1_r3batch.opts
- FILEOPTS_DYN_POOL_r3batch.opts

AGENT2

- FILEOPTS_AGENT2_r3batch.opts
- FILEOPTS_DYN_POOL_r3batch.opts

Extended agents

All access methods use two types of options file:

Global options file

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

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

For PeopleSoft

psagent.opts

For SAP

r3batch.opts

For custom access methods

netmth.opts

Local options file

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

XA_Unique_ID

The unique identifier of the workstation in the plan. Since the folder feature was introduced, workstations can be defined in folders, therefore, the workstation name alone, is not sufficient to identify a workstation in the plan, but instead, the name and folder combination is mapped to a unique identifier. In the localopts file, the value of the this_cpu option is the unique identifier of the workstation.

You can also verify the unique identifier for a workstation by submitting the composer list command with the ;showid filter. Should this result in blank, *XA_Unique_ID* corresponds to the workstation name. You can also retrieve the unique identifier submitting the conman showcpus command with the ;showid filter. For example, if the installation of the `r3batch` access method includes two extended agent workstations, with unique identifiers `S4HANAR3BW` and `07756YBX76Z6AFX2`, then the names of the local options files are `S4HANAR3BW_r3batch.opts` and `07756YBX76Z6AFX2_r3batch.opts`.

accessmethod

Is the name of the access method.

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

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

PeopleSoft

Configuring the PeopleSoft access method on page 33.

SAP

Configuring the SAP access method on page 77.

Option value inheritance

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

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

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

Defining supported agent workstations

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

Creating a workstation using the Dynamic Workload Console

About this task

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

Dynamic agents

The agents are automatically registered to the IBM Workload Scheduler network.

Extended agents

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

1. From the Dynamic Workload Console portfolio, click **Administration > Workload Environment Design > Create Workstations**.
2. Select an engine from the list and click **Create Workstation**.
3. In the Workstations properties panel, specify the attributes for the extended agent workstation you are creating. For all the details about available fields and options, see the online help by clicking the "?" in the top-right corner. In the workstation definition, specify the access method and other properties, as shown in Table 1: How to complete the extended agents definition on page 23. For further information about the workstation definition properties, see the section about workstation definition in *IBM Workload Scheduler User's Guide and Reference*.
4. To assign the workstation to an existing domain or to create a new domain, click **Assign to Domain**.
5. Click **Save**.

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

Table 1. How to complete the extended agents definition

Field	Description by Access Method	
	PeopleSoft	**SAP**
Name	The name for the extended agent workstation. For all access methods the name must start with a letter and can contain alphanumeric characters, dashes, and underscores. The maximum length is 16 characters. Workstation names must be unique and cannot be the same as workstation class and domain names. Double-byte character set (DBCS), single-byte character set (SBCS), and bidirectional text are not supported. If a workstation name contains these characters and, as a result, the options file name contains the same name, the workstation cannot be validated by the SAP system.	

Table 1. How to complete the extended agents definition

(continued)

Field	Description by Access Method	
	PeopleSoft	**SAP**
	For all the access methods, the name of the options file associated with the extended agent workstation must contain the unique identifier of the extended agent workstation, *XA_Unique_ID_*<access-method>.opts. That is, if the unique identifier for the extended agent workstation named /SAPBUS/S4HANAR3BW is 07756YBX76Z6AFX2, then the options file name must be 07756YBX76Z6AFX2_-r3batch.opts. For information about retrieving the unique identifier for an extended agent workstation, see UNIQUE_ID on page 21.	
TCP Port	Any number other than 0.	
Access Method	**psagent**	**r3batch** **Note:** In UNIX™ the name is case sensitive and must be lowercase.

Creating a workstation using the command line

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

Dynamic agents

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

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

Extended agents

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

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

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

Defining workstations for end-to-end scheduling

About this task

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

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

Extended agents

Extended agent workstations must be defined as fault-tolerant workstations in IBM Z Workload Scheduler.

A fault-tolerant workstation is the IBM Z Workload Scheduler definition of an existing IBM Workload Scheduler agent in the distributed network. The IBM Workload Scheduler agent is where the job associated with the fault-tolerant workstation actually runs in the distributed network.

To define the extended agent workstation in IBM Z Workload Scheduler, you must:

1. Define the workstation in the `CPUREC` initialization statement. For an example, see Creating the CPUREC statement for extended agents on page 25.
2. Add the same workstation definition to the database using ISPF or the Dynamic Workload Console. For a description of how to define the workstation using the Dynamic Workload Console, see *Dynamic Workload Console User's Guide*. For an example, see Defining the workstation with ISPF on page 27.

IBM Z Workload Scheduler agents

To define the agent workstation with z-centric capability in IBM Z Workload Scheduler, add the workstation definition to the database using ISPF or the Dynamic Workload Console. For further information, see *Scheduling End-to-end with z-centric Capabilities*.

Creating the CPUREC statement for extended agents

This section is valid only for Extended agents. Create the `CPUREC` statement for the workstation in the `TOPOLOGY` initialization statement. The `TOPOLOGY` initialization statement is used to define parameters related to the topology of the connected IBM

Workload Scheduler network. Such a network topology statement is made up of one or more (one for each domain) `DOMREC` statements that describe the topology of the distributed network, and by several `CPUREC` statements, one for each fault-tolerant workstation.

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

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

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

The following keywords define `R3XA` as an extended agent:

CPUACCESS

The extended agent access method. For SAP, it is `r3batch`.

CPUHOST

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

CPUTYPE

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

Note: The `CPUREC` statement does not exist for an IBM Workload Scheduler for z/OS agent workstation.

For further information about `CPUREC` for extended agents, see *Customization and Tuning*.

Defining the workstation with ISPF

About this task

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

Extended agents

In ISPF, define the workstation as `computer automatic` and then set the *FT Work station* field to **Y**. The `CPUREC` statement with the three keywords described in Creating the CPUREC statement for extended agents on page 25 provides the extended agent specification.

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

Figure 1. Defining an Extended Agent workstation

```
-------------- CREATING GENERAL INFORMATION ABOUT A W0RK STATION --------------
 Command ===>

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

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

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

IBM Z Workload Scheduler agents

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

Defining jobs for supported agents

To run and monitor a PeopleSoft, SAP, or z/OS job with IBM Workload Scheduler, the supported agents, or access method require an IBM Workload Scheduler job definition, where you specify the external job you want to schedule, the workstation

(also defined in IBM Workload Scheduler) on which it is to run, and any recovery actions. To define the job, use either of the following methods:

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

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

- Dynamic Workload Console.
- IBM Z Workload Scheduler ISPF dialogs. You must also create a member in the `SCRIPTLIB` with a `JOBREC` statement for the job.

Jobs defined for supported agents are added to job streams and scheduled in the same way as any other job in IBM Workload Scheduler and IBM Z Workload Scheduler.

Defining jobs with the Dynamic Workload Console

About this task

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

Steps for defining a job for supported agents.

To define jobs, follow these steps:

1. From the Dynamic Workload Console, click the **Design** menu and select Workload Designer.
2. Specify an engine name, either distributed or z/OS. The Workload Designer window opens. Job types and characteristics vary depending on whether you select a distributed or a z/OS engine.
3. Click **Create new** and select **Job definition**.
4. Select the category and type of job you want to create.
    - For SAP jobs, **ERP > SAP Job on XA Workstations** or **SAP Job on Dynamic Workstations**. See Creating SAP Standard R/3 jobs from the Dynamic Workload Console on page 102.
    - For z/OS and PeopleSoft, **ERP > Access Method**.
5. In the properties panel, specify the attributes for the job definition you are creating. For all the details about available fields and options, see the online help specific for the item.
6. Click **Save** to save the job definition in the database.

Note: The access method for SAP provides supplementary features if you use the alternative steps described in Creating SAP Standard R/3 jobs from the Dynamic Workload Console on page 102. You can create native SAP Standard jobs on a remote SAP system directly from the Dynamic Workload Console.

Defining jobs using the command line

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

Dynamic agents

The following example describes an IBM Workload Scheduler job named `DYN_JOB_R3_0001` defined in the folder name `SAPJOBS`, that runs on a dynamic agent workstation named `NC112015_1`. The IBM Workload Scheduler launches a job in an SAP environment named `JOB_APPS_93`.

```
NC112015_1#/SAPJOBS/DYN_JOB_R3_0001
 TASK
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition
     xmlns:jsdl="http://www.abc.com/xmlns/prod/scheduling/1.0/jsdl"
     xmlns:jsdlxa="http://www.abc.com/xmlns/prod/scheduling/1.0/jsdlxa"
                  name="r3">
  <jsdl:application name="r3" plugin="xajob">
    <jsdlxa:xajob accessMethod="r3batch" target="NW73LIN_r3batch">
            <jsdlxa:taskString>/ -job JOB_APPS_93 -i 14514200 -c c
                 -flag ENABLE_APPL_RC </jsdlxa:taskString>
        </jsdlxa:xajob>
  </jsdl:application>
</jsdl:jobDefinition>
 RECOVERY STOP
```

Extended agents

The following example describes an IBM Workload Scheduler job named `psjob2` that runs on a PeopleSoft extended agent workstation with unique identifier named `XAPS002`. IBM Workload Scheduler logs on to UNIX operating system as `psjobs` and launches a job under PeopleSoft. The PeopleSoft process is named `XRFWIN`. If recovery is needed, IBM Workload Scheduler runs job `recov2` and then continues processing.

```
XAPS002#/mypsjobs/psjob2
  streamlogon psjobs
  scriptname
   -process XRFWIN -type 'SQR Report' -runcontrol 1 -runlocationdescr PSNT
  description "peoplesoft job #2"
  recovery continue after recov2
```

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

- Task string parameters for PeopleSoft jobs on page 42.
- Task string to define SAP jobs on page 110.
- Task definition syntax for z/OS jobs scheduled with IBM Workload Scheduler.

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

Defining jobs for end-to-end scheduling

Extended agents

Extended agent jobs scheduled to run in an end-to-end environment cannot be defined using the Dynamic Workload Console or the IBM Workload Scheduler command line, but must be added to the `SCRIPTLIB` of IBM Z Workload Scheduler.

In the `OPERATIONS` ISPF panel of IBM Z Workload Scheduler, extended agent jobs are defined like any other job, but with the specific attribute for a job defined on an extended agent workstation. The following example

shows the definition of a job named `SAPJOB`. This is the IBM Z Workload Scheduler job that drives the running of on SAP R/3 job (named `BAPRINT46B` as shown in the next example). It shows as an extended agent job because the associated workstation is an extended agent workstation named `R3XA`.

Figure 2. Defining an Extended Agent job for end-to-end scheduling

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

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

Application            : APLL1            FTW appl

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

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

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

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

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

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

IBM Z Workload Scheduler agents

For information about the jobs definition for agent with z-centric capabilities, see *Scheduling End-to-end with z-centric capabilities*.

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

- Task string parameters for PeopleSoft jobs on page 42.
- Task string to define SAP jobs on page 110 or Defining SAP jobs dynamically on page 128.
- Task definition syntax for z/OS jobs scheduled with IBM Workload Scheduler.

Submitting jobs

About this task

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

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

   Dynamic agents

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

   Extended agents

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

   IBM Z Workload Scheduler agents

   - *IBM Z Workload Scheduler: Scheduling End-to-end with z-centric Capabilities* for **ISPF** panel.
   - *Dynamic Workload Console User's Guide* for Dynamic Workload Console.

Chapter 6. Access method for PeopleSoft

What you need and what you can do with Access method for PeopleSoft.

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

For information about the supported versions of the plug-ins and access methods, run the Data Integration report and select the **Supported Software** tab.

Features

Look at the tasks you can perform by using Access method for PeopleSoft.

Using Access method for PeopleSoft, you can perform the following tasks:

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

Roles and responsibilities

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

In a typical enterprise, different users contribute to the implementation and operation of the product. Table 2: Roles and responsibilities in Access method for PeopleSoft on page 32 describes the roles and responsibilities of all those involved in the process model, showing the tasks they perform.

Table 2. Roles and responsibilities in Access method for PeopleSoft

User role	User task
IBM Workload Scheduler configurator	Defining the configuration options on page 34
IBM Workload Scheduler developer	• Defining PeopleSoft jobs in IBM Workload Scheduler on page 41 • Configuring the job status mapping policy on page 44

Table 2. Roles and responsibilities in Access method for PeopleSoft (continued)

User role	User task
PeopleSoft administrator	• Creating a batch processing ID in PeopleSoft on page 37 • Configuring the ITWS_PSXA PeopleSoft project on page 38 • Uploading the PeopleSoft project on page 38

Scheduling process for the PeopleSoft supported agents

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

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

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

PeopleSoft job tracking in IBM Workload Scheduler

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

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

Security

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

Configuring the PeopleSoft access method

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

Defining the configuration options

The IBM Workload Scheduler installation process creates a default global options file for the `psagent` access method, named `psagent.opts`. You can also create the following local files in the path:

On UNIX operating systems

psjoa.jar

```
TWA_DATA_DIR/methods
```

On Windows operating systems

```
TWA_home\methods
```

Extended agent

`XA_Unique_ID_psagent.opts` where `XA_Unique_ID` is the unique identifier for the extended agent workstation. For more details about how to identify the unique ID, see UNIQUE_ID on page 21.

Dynamic agent

`DYNAMIC_AGENT_FILE_psagent.opts` where *DYNAMIC_AGENT_FILE* is any text string. This string does not necessarily correspond to the name of the dynamic agent workstation since the dynamic agent can have more than one .opts file associated. For more information, see Setting options for the access methods on page 19.

To edit both options file, you can use any text editor. On dynamic workstations, you can edit the options files from the job definition panels in the Dynamic Workload Console. For examples of options files for this access method, see PeopleSoft options file example on page 36.

Table 3: Psagent access method options on page 34 describes the options for the `psagent` access method. Option names are case insensitive. Before you use a manually-created options file, check that all the option names are written correctly, otherwise they will be ignored.

Table 3. Psagent access method options

Table 3. Psagent access method options (continued)

Option	Description
	For details about how to encrypt the password, see Encrypting PeopleSoft operator passwords on page 37.
PSFT_OPERATOR_ID	(Mandatory) Specifies the PeopleSoft operator ID used for the connection to the PeopleSoft application server.
PSFT_OPERATOR_PWD	(Mandatory) Specifies the encrypted password (case-sensitive) of the PeopleSoft operator ID used for the connection to the PeopleSoft application server. For details about how to encrypt the password, see Encrypting PeopleSoft operator passwords on page 37.
PSJOAPATH	(Optional) Specifies the full path name of the `psjoa.jar` file, containing both the path and the psjoa.jar filename. If this option is not set, the following default path name is used: **UNIX** *TWA_DATA_DIR*`/methods/psjoa.jar` **Windows** *TWA_home*`\methods\psjoa.jar` Ensure that you identify the version of the `psjoa.jar` file that corresponds to the version of PeopleSoft that you are using and you have access to the file.
RUNLOCATION	(Optional) Specifies the default PeopleTools process server that processes the requests.
SERVER_NAME_LIST	(Mandatory) Specifies the list of application servers that the psagent tries to connect to. It is a comma-separated list of addresses in the format: *server*`:`*port* `[,`*server*`:`*port*`] ...` where: ***server*** Specifies the host name or TCP/IP address of the server ***port*** Specifies the port number the server is listening on.
TWS_MAX_WAIT_TIME	(Optional) Specifies the maximum time that the supported agent waits (timeout) after a failed operation on the PeopleSoft application server before retrying the operation. The default is 10 seconds.

Table 3. Psagent access method options (continued)

Option	Description
TWS_MIN_WAIT_TIME	(Optional) Specifies the minimum time that the supported agent waits (timeout) after a failed operation on the PeopleSoft application server before retrying the operation. The default is 5 seconds.
TWS_RETRY	(Optional) The maximum number of times that the supported agent attempts to re-run a failed operation on the PeopleSoft application server. The default is 5.
TWSXA_INLINE_CI	(Optional) Specifies the name of the component interface that the `psagent` invokes to submit jobs to PeopleSoft. The default is `ITWS_PROCESSREQUEST`. If you use this default, you must perform the customization steps described in Configuring the ITWS_PSXA PeopleSoft project on page 38. If you do not plan to schedule jobs containing in-line variables, and you do not want to perform the additional customization steps, you must replace the default value with `PROCESSREQUEST`. This is the component interface invoked by previous versions of the access method; it does not allow the use of in-line variables.
TWSXA_SCHED_METH	(Optional) Specifies the name of the PeopleSoft method invoked by the component interface specified in `TWSXA_INLINE_CI`. Both `ITWS_PROCESSREQUEST` and `PROCESSREQUEST` use the default method `Schedule`. If you are using either of these component interfaces, leave the default. If you are using a different component interface, specify the name of the method called by your component interface, respecting the case of the PeopleSoft object name.

PeopleSoft options file example

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

Remember to save the file in the following directory:

UNIX

```
TWA_DATA_DIR/methods
```

Windows

```
TWA_home\methods
```

Example

```
LJuser=TwsUsr
CheckInterval=120
```

```
PSFT_OPERATOR_ID=PSHC
PSFT_OPERATOR_PWD=*****
SERVER_NAME_LIST=9.87.120.36:9000
```

If you create the options file manually, you must encrypt the PeopleSoft operator password, as described in Encrypting PeopleSoft operator passwords on page 37.

Encrypting PeopleSoft operator passwords

When you add or change the PeopleSoft operator password using the Dynamic Workload Console, the password is automatically encrypted and securely stored in the file. For added security, it is displayed on the screen as a series of asterisks.

When you add or change the PeopleSoft user password using a text editor, run the **pwdcrypt** command to encrypt the password before writing it in the file, as follows:

```
pwdcrypt password
```

The program returns the password in encrypted format that you can then copy and paste into the options file.

Connecting to more than one PeopleSoft application server

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

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

```
PS847SYSTEM_psagent.opts
```

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

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

Note: In case you specify some connection properties in your local option files, make sure that the same properties are commented out in your global option file, with the exception of the global property **LJuser**. This action is needed to avoid that warning messages related to duplicate properties are displayed in the job log.

Creating a batch processing ID in PeopleSoft

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

Configuring the ITWS_PSXA PeopleSoft project

About this task

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

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

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

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

To identify work records, use the following PeopleSoft naming conventions:

- A derived work record name ends with '_WRK'.
- A work record definition name for Structured Query Report reports starts with R_

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

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

- Leave the value of the `TWSXA_INLINE_CI` option set to `ITWS_PROCESSREQUEST`, that is the default value. See Defining the configuration options on page 34 for a detailed explanation.
- Upload the PeopleSoft project as described in Uploading the PeopleSoft project on page 38.

Uploading the PeopleSoft project

About this task

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

After installing the product, complete the following steps:

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

   UNIX

   ```
   TWS_DIR/methods/
   /---PeopleSoft
           /---PT844
            /---ITWS_PSXA
                    ITWS_PSXA.ini
                    ITWS_PSXA.XML
   ```

   Windows

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

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

   The project is contained in the `TWS_DIR/methods/PeopleSoft/PT844` subdirectories on UNIX® and in `TWA_home\methods\PeopleSoft\PT844`. on Windows™.

   After you specify the project folder, a list of projects appears in the **Project Name** field of the Copy Project From File window.

5. Choose `ITWS_PSXA` and click **Open**. If you already configured `ITWS_PSXA` (perhaps after installing a fix pack), a confirmation window enquires if you want to replace the existing one. Click **Yes**.

   The Copy window is displayed showing a list of definition types.

6. Click **Options** to select the new settings.
   a. Click **Report Filter**
   b. Click **Select All**
   c. Click **OK**

d. Click **Select All**
e. Click **Copy**. A progress bar is displayed.

After loading the project, the PeopleSoft Database contains the following objects:

- ITWS process type definitions
- ITWS permissions list
- ITWS component interfaces

7. Create the `ITWS_ROLE` security role. You can use either the PeopleSoft Web GUI or the Application Designer. Follow the steps below:

   From the menu of the PeopleSoft Web GUI:

   a. Select **NavBar** in the upper right corner -> **Menu** -> **PeopleTools** -> **Security** -> **Permission and Roles** -> **Roles**.
   b. Select the **Add a new value** tab
   c. Type or select `ITWS_ROLE` in the **Role Name** field
   d. Select the **Permissions list** tab -> **ITWS** -> **Save**

   From the Application Designer GUI:

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

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

   From the PeopleSoft Web GUI:

   a. Select **NavBar** in the upper right corner -> **Menu** -> **PeopleTools** -> **Security** -> **User Profiles**.
   b. Type the user name of the user who wants to schedule jobs from IBM Workload Scheduler.
   c. In **User Roles** window click on the **+** sign in the row where you want to add the role.
   d. Select the **Roles** tab.
   e. Add `ITWS_ROLE` and save.

   From the Application Designer GUI:

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

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

   From the PeopleSoft Web GUI:

   a. Select **NavBar** in the upper right corner -> **Menu** -> **PeopleTools** -> **Process Scheduler** -> **Process Scheduler Servers**.
   b. Click on the **Search** button and select the PeopleTools server you plan to use.
   c. In the **Server Definition** tab and in **Process Types run on this Server** window click on the **+** sign in the row of your choice.
   d. Click on the search icon. A new windows is displayed.

e. Select all the **ITWS_*** entries (one for each row created).
f. Click Save.

From the Application Designer GUI:

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

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

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

10. Restart the process servers.

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

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

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

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

Defining PeopleSoft jobs

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

Defining PeopleSoft jobs in IBM Workload Scheduler

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

- The name of the IBM Workload Scheduler job that runs the PeopleSoft job
- The unique identifier of the extended agent or dynamic workstation or workstation class where the IBM Workload Scheduler job runs. See UNIQUE_ID on page 21 for more information about the unique identifier.
- The name of the user launching the job
- Recovery options
- The Script file specifications

For more information, refer to Defining jobs for supported agents on page 27.

Task string parameters for PeopleSoft jobs

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

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

  The following is an example of a `JOBREC` statement:

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

  where:

  process_name

  The process name for the PeopleSoft job.

  process_type

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

  runcontrol_ID

  The runcontrol ID for the PeopleSoft job.

  TWS_user_name

  The IBM Z Workload Scheduler user who runs the `psagent` access method from the end-to-end scheduling environment.

Table 4: Task string parameters for PeopleSoft jobs on page 42 describes the parameters to define PeopleSoft jobs.

Table 4. Task string parameters for PeopleSoft jobs

Parameter	Description
-process	The process name for the PeopleSoft job.
-type	The process type for the PeopleSoft job. This entry must be enclosed within single quotes.
-runcontrol	The runcontrol ID for the PeopleSoft job.
-outputdest	The destination of the PeopleSoft job output.
-outputtype	The output type of the PeopleSoft job. Possible values are:

Table 4. Task string parameters for PeopleSoft jobs (continued)

Parameter	Description
	• Any • Email • File • None • Printer • Web • Window If you do not specify any value, IBM Workload Scheduler uses the value associated to the PeopleSoft job you are submitting. **Note:** Depending on the PeopleSoft configuration, some combinations of the value of this option with the value of the **outputformat** option are not supported. In this case the PeopleSoft default value is used.
-outputformat	The output format of the PeopleSoft job. Valid values are: **None** PDF **CSV** PS **DOC** RPT **Default** RTF **HTM** TXT **LP** WKS **OTHER** XLS

Table 4. Task string parameters for PeopleSoft jobs (continued)

Parameter	Description
	Note: Depending on the PeopleSoft configuration, some combinations of the value of this option with the value of the **outputtype** option are not supported. In this case the PeopleSoft default value is used.
-runlocationdescr	The PeopleSoft process scheduler responsible for processing the PeopleSoft job.
-foldername	The name of the report folder used for this job. The folder must have been already created using PeopleSoft Report Manager.
tracelvl	Specify the trace setting for the job. Possible values are: **1** Only error messages are written in the trace file. This is the default. **2** Informational messages and warnings are also written in the trace file. **3** A most verbose debug output is written in the trace file. Refer to Configuring the tracing utility on page 54 for detailed information.

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

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

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

Configuring the job status mapping policy

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

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

0

The distribution status is ignored and the IBM Workload Scheduler job status is calculated as shown in Table 6: Relationship between the run status and the IBM Workload Scheduler job status on page 46.

1

The distribution status is used and the IBM Workload Scheduler job status is calculated as shown in Table 5: Relationship between the run status, the distribution status, and the IBM Workload Scheduler job status on page 45. This is the default value.

Table 5: Relationship between the run status, the distribution status, and the IBM Workload Scheduler job status on page 45 shows the relationship between the run status, the distribution status, and the IBM Workload Scheduler job status. The return code associated with the status is shown in parentheses. IBM Workload Scheduler uses this return code to evaluate the return code condition you specified in the **Return Code Mapping Expression** field in the Properties panel of the job definition. For more details about this field, refer to the online help by clicking the "?" in the top-right corner of the panel.

Table 5. Relationship between the run status, the distribution status, and the IBM Workload Scheduler job status

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

Table 6: Relationship between the run status and the IBM Workload Scheduler job status on page 46 shows the relationship between the PeopleSoft run status and the IBM Workload Scheduler job status. The return code associated with the status is shown in parentheses. IBM Workload Scheduler uses this return code to evaluate the return code condition you

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

Table 6. Relationship between the run status and the IBM Workload Scheduler job status

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

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

Chapter 7. Common serviceability for the access methods

This section provides information common to all the access methods including return code mapping, configuring the tracing utility, and troubleshooting the access method.

The return code mapping feature

The return code mapping feature provides a standard way of mapping messages into return code values. You can also customize the return code mapping. This feature is available for the following access methods:

- PeopleSoft
- SAP

The return code mapping feature provides more granularity when defining the success or failure policies of jobs and improved flexibility in controlling job execution flows based on execution results. Job return code mapping provides the following capabilities:

- Users can define a job final status (successful or failed) based on a condition on the return code of the execution of the program or script of the job.
- The return code can be provided also to the recovery job that is associated with it in the job definition. This causes the recovery job to perform different processing based on the return code.

Parameters

#

Optional comment. All the lines starting with this symbol (#) are not used for mapping.

pattern*n*

Pattern strings delimited by quotation marks (“ and ”). If you use only one pattern string, you can omit the quotation marks. If the pattern string contains a quotation marks character, then it must be escaped by backslash (\). The string can contain the following wildcards and special characters:

Asterisk (*)

Matches an arbitrary number of characters.

Question mark (?)

Matches a single character.

Backslash (\)

Escape character.

RC value

The return code value. This value is sent by the method to IBM Workload Scheduler by a `%RC nnnn` message.

Creating a return code mapping file

You can create a return code mapping file to customize your own return codes with respect to certain conditions that might affect a job when it runs. Use this file to set the success condition of the job, which IBM Workload Scheduler uses to assess if the job completes successfully or in error. The return code is sent to IBM Workload Scheduler in the form of a `%RC nnnn` message. If this message is received, the job state is updated accordingly.

Each method has its own set of files to map the messages into return code values. The mapping files can be either global or local for a workstation.

Return code mapping files that are specific to a workstation are named according to the following scheme:

UNIX

TWA_DATA_DIR`/methods/rcm/`*accessmethod-type-workstation*`.rcm`

Windows

TWA_home`\methods\rcm\`*accessmethod-type-workstation*`.rcm`

Global mapping files have a file name according to the following scheme:

UNIX

TWA_DATA_DIR`/methods/rcm/`*accessmethod-type*`.rcm`

Windows

TWA_home`\methods\rcm\`*accessmethod-type*`.rcm`

For the PeopleSoft access method, *type* is always equal to `rcmap`. For the SAP R/3 access method, *type* is as described in Return code mapping file names for r3batch on page 51.

Syntax

About this task

Use the following syntax to create the return code mapping file:

```
[#] “pattern1? ”pattern2”…”patternn” = RC value
```

Examples

The following is an example of a return code mapping file. The line numbers in bold do not belong to the file, but are shown for reference:

```
1.   # This is an RC mapping file for joblog.
2.
3.   “User * missing  = 102
4.   “\*\*\*?  = 103
5.   “User \
6.   * \
7.   missing  = 102
```

In this example:

- Line **1** is a comment and is not used for mapping.
- Line **2** is blank and is ignored. All blanks preceding or following a pattern string are ignored, as well as those between the equals sign and the return code value.
- Line **3** matches every message starting with the string *User* and ending with the string *missing*.
- Line **4** matches every message starting with three asterisks (*) followed by a blank. When you use the asterisk in this way and not as a wildcard, you must escape it with a backslash.
- Lines **5** through **7** contain a pattern taking several lines. It matches the same messages as the pattern of line **3**.

Considerations

Note the following facts:

- The order of the pattern lines is important because the first matching pattern line is used to build the return code value.
- Empty pattern strings (“”) are ignored by the pattern matching procedure.

For example, the following is a valid pattern sequence. The first line is more restrictive than the second line.

```
“625 “User * missing = 104
“ “User * missing = 102
```

The following pattern sequence is formally valid, but the second pattern line is never used. Because the first line is more general, it is always matched first.

```
“ “User * missing = 102
“625 “User * missing = 104
```

Return code mapping for psagent

For the PeopleSoft access method, you can write return code mapping files associating the internal states listed in Table 7: Job states and return codes for the PeopleSoft access method on page 49.

When no return code mapping files are defined, or when a string returned by the access method does not satisfy any of the matching patterns of the mapping file, the access method uses the respective standard return codes listed in the tables.

Table 7. Job states and return codes for the PeopleSoft access method

psagent job state	psagent return code
"CANCEL"	1
"DELETE"	2
"ERROR"	3
"HOLD"	4
"QUEUED"	5

Table 7. Job states and return codes for the PeopleSoft access method (continued)

psagent job state	psagent return code
"INITIATED"	6
"PROCESSING"	7
"CANCELED"	8
"SUCCESS"	9
"NO SUCCESSPOSTED"	10
"POSTED"	11
"NOT POSTED"	12
"RESEND"	13
"POSTING"	14
"GENERATED"	15

Return code mapping for r3batch

About this task

Using return code mapping with `r3batch` can be useful in overcoming differences in the return code mechanisms of R/3, which returns a mixture of messages and numbers, and of IBM Workload Scheduler, which handles exclusively numeric return codes. By customizing the return code mapping files listed in Return code mapping file names for r3batch on page 51, you can map messages from R/3 logs, spool lists, and exceptions from RFC function modules into return code values that IBM Workload Scheduler can handle.

Note that when you do not use this feature, `r3batch` does not send any return codes to IBM Workload Scheduler. In this case, IBM Workload Scheduler displays only the `r3batch` exit code, which cannot be used to set up `rccondsucc` conditions.

The return code mapping mechanism works as follows:

1. `r3batch` reads the output retrieved from the R/3 system (R/3 job log, process chain log, spool list, and so on appended to the `stdlist` of the related IBM Workload Scheduler job).
2. Following your specifications in the `rcm` files, the R/3 return messages or codes are mapped into your custom return codes and passed on to IBM Workload Scheduler.
3. These return codes are used together with the `rccondsucc` keyword set in the extended agent job definition and handled accordingly. Return code mapping is meaningful only if you use the return codes to write the expressions that determine job completion. Conversely, in the case of this extended agent, the use of `rccondsucc` is significant only if IBM Workload Scheduler gets return codes (not exit codes) from the access method.

To use the return code mapping feature:

- Leave the value of the `rcmap` option as `ON` (this is the default).
- Depending on which R/3 logs you want `r3batch` to read and map, leave the default settings of the `retrieve_joblog`, `retrieve_pchainlog`, and `retrieve_spoollist` options as `ON` and manually create the corresponding `rcm` files.
- If you want to map messages from the R/3 syslog, set the `log_r3syslog` option to `ON` and manually create the corresponding `rcm` file.

When setting up your return code mapping for `r3batch`, consider the following:

- You can define any return code numbers for your use because there are no reserved return codes for the access method or for IBM Workload Scheduler.
- Mapping files are scanned sequentially: the first match found performs the corresponding mapping. When you define a mapping file, write the most restrictive strings first.
- When you define a mapping file, remember that the R/3 log messages are read in their entirety. If you want to map only a part of the entry, you must use the wildcard characters.
- If two lines match two different patterns, then the return code is set to the higher value. In general the return code is set to the highest value among the ones yielded by the matched patterns. This is shown in the following example:

  The job log returned after job PAYT410 has run is:

  ```
  *** ERROR 778 ***  EEWO0778E Failed to modify the job PAYT410 with job id
  ***       05710310.

  *** ERROR 176 ***  EEWO0176E Failed to add step 1.

  *** ERROR 552 ***  EEWO0552E The R/3 job scheduling system has found an
  ***       error for user name * and job name PAYT410. Please check R/3
  ***       syslog.

  *** ERROR 118 ***  EEWO0118E Execution terminated. Could not create and
  ***       start an instance of the R/3 batch job.
  ERROR LEVEL=118
  ```

  and the system log contains the following line:

  ```
  |011:05:12|MAESTRO|SAPMSSY1|EFT|> Step 1 contains illegal values
  ```

  The `r3batch-joblog.rcm` file contains the following matching line:

  ```
  "118""*"=100
  ```

  while the `r3batch-syslog.rcm` file contains the following matching line:

  ```
  "*MAESTRO*Step 1 contains illegal values"=9999
  ```

  In this case, the return code sent back to IBM Workload Scheduler is 9999 because it is the higher of the two matching patterns.
- If no matching takes place, no return code is sent to IBM Workload Scheduler.

Return code mapping file names for r3batch

On UNIX operating systems

TWA_DATA_DIR/methods

On Windows operating systems

TWA_home\methods

You can create the mapping files you want to implement in the `rcm` directory:

r3batch-joblog.rcm

Maps messages from the R/3 job log of a job into return code values. If this file is not present, the messages in the job log are ignored.

The format of the mapping file is:

```
 message_text_pattern
[program_pattern[message_number_pattern[message_id_pattern]]]=RCvalue
```

where *program_pattern* is the external program that produced the output shown in the job log and *message_id_pattern* is the message class. For example, the following line appended in the job log:

```
04/26/2005 10:08:04 00
    550Step 001 started (program BTCTEST, variant VAR1, user name TWSDEV)
```

will match the following pattern line in `r3batch-joblog.rcm`:

```
"*Step*" "*" "550" "*"=5
```

because:

message_text_pattern

`"Step 001 started (program BTCTEST, variant VAR1, user name TWSDEV)"`

program_pattern

`"*"`

message_number_pattern

`"550"`

message_id_pattern

`"*"`

r3batch-pchainlog.rcm

Maps messages from the protocol of a Process Chain into return code values. If this file is not present, the messages in the protocol are ignored.

The format of the mapping file is:

```
 message_number_pattern
[message_id_pattern[message_variable1[message_variable2
[message_variable3[message_variable4[message_type]]]]]]=RCvalue
```

`r3batch-spoollist.rcm`

Maps messages in the job spool list of an R/3 job into return code values. If this file is not present, the messages in the spool list are ignored.

The format of the mapping file is:

```
spool_list_row_pattern=RCvalue
```

`r3batch-syslog.rcm`

Maps messages in the syslog of an R/3 system into return code values. The R/3 system log should be checked only when R/3 returns the generic 552 error to `r3batch`.

If this file is not present, the messages in the system log are ignored.

The format of the mapping file is:

```
system_log_row_pattern=RCvalue
```

If you plan to map system log messages, be sure to set the `log_r3syslog` option of `r3batch` to `ON` (the default is `OFF`).

`r3batch-msgrc.rcm`

Maps ABAP exceptions and BAPI return codes of RFC function modules into return code values. If this file is not present, the mapping is done using a hardcoded table.

The format of the mapping file is:

```
message_number=RCvalue
```

`message_number` is the error message number. The last message number is always used. That is, if two error messages are generated, only the second one is checked against the mapping file.

Mapping return codes for intercepted jobs

About this task

To set up return code mapping for intercepted jobs, after defining the appropriate return code conditions in the `r3batch-joblog.rcm` file, do the following:

1. Create a customized template file named `rctemplate.jdf` in the following directory:

   UNIX

   TWA_DATA_DIR`/methods/r3batch_icp/`

   Windows

   ```
   TWA_home\methods\r3batch_icp\
   ```

   The file must contain the following:

   ```
   alias;rccondsucc "Success Condition"
   ```

where, the "Success Condition" must match a condition saved in the `rcm` file.

2. Modify the *XANAME*`_r3batch.icp` file, located in the same path, to refer to the `jdf` file you created in the previous step as follows:

   ```
   client job_mask user_mask rctemplate
   ```

   IBM Workload Scheduler manages the intercepted R/3 job as a docommand job with all the options specified in the customized `jdf` file. You can check if your intercepted job is correctly submitted by reading the job_interceptor joblog.

Configuring the tracing utility

Learn how to configure the trace utility for all the access methods.

IBM Workload Scheduler logs all the processing information in the following configuration file:

UNIX

TWA_DATA_DIR`/methods/`*accessmethod*`.properties`

Windows

```
TWA_home\methods\accessmethod.properties
```

Note: If you delete this file accidentally, IBM Workload Scheduler creates a new file with all the default values and contains the following comment:

```
# This file was automatically created using the default values.
```

Customizing the .properties file

About this task

Depending on the access method you are working with, customize the trace parameters in the following properties files:

psagent.properties

For the PeopleSoft access method.

r3batch.properties, r3evmon.properties

For the SAP access method.

With this access method, you can also specify debug and trace parameters in the single job definitions. See Creating SAP Standard R/3 jobs from the Dynamic Workload Console on page 102 and Task string to define SAP jobs on page 110.

For each .properties file you can customize the following parameters:

***accessmethod*.trace.tracers.level**

Specify the level of tracing you want to set. Possible values are:

DEBUG_MIN

Only error messages are written in the trace file. This is the default.

DEBUG_MID

Informational messages and warnings are also written in the trace file.

DEBUG_MAX

A most verbose debug output is written in the trace file.

The value you set in the .properties file applies to all the jobs of the corresponding access method. To specify a different trace setting for a particular job, specify the following option in the job definition:

```
-tracelvl=(1|2|3)
```

where:

- `1 = DEBUG_MIN`
- `2 = DEBUG_MID`
- `3 = DEBUG_MAX`

Note: When making changes to the trace level setting, the changes are effective immediately after saving the .properties file. No restart is required.

***accessmethod*.trace.handlers.traceFile.fileDir**

Specifies the path where the trace file is created. Depending on the access method, the default is:

SAP

On UNIX® operating systems

TWA_DATA_DIR/methods/traces

On Windows™ operating systems

TWA_home\methods\traces

All other access methods

On UNIX operating systems

TWA_DATA_DIR/methods

On Windows operating systems

TWA_home\methods

Ensure that the new path you specify has already been created as a fully qualified path with write permissions.

Traces are written in XML format. Always use slashes (or backslashes) when you specify a new path, even if you are working on Windows™ operating systems.

The trace files give information about the method execution to the desired level of detail. The minimum trace level is always on, to guarantee a First-Failure Data Capture (FFDC) ability. The trace file name is:

trace-psagent.log

For the PeopleSoft access method.

trace-r3batch.log, trace-*XAname*-r3evmon.log

For the SAP access method.

***accessmethod*.trace.tracers.logging**

Specifies to enable or disable the trace utility. Possible values are:

true

To enable the trace utility. This is the default value.

false

To disable the trace utility. If you set this parameter to **false**, no traces are written in the trace-*accessmethod*.log file even if there are problems.

r3batch.trace.handlers.traceFile.maxFiles

The maximum number of trace files that are created before the oldest one is deleted. If this parameter is set to 1, the current trace file is never replaced and can grow without limit.

r3batch.trace.handlers.traceFile.maxFileSize

The maximum size (in bytes) that the trace file can reach before it is renamed and a new trace file is created. This parameter is valid only if the r3batch.trace.handlers.traceFile.maxFiles is set to a value greater than 1.

Configuration file example for the SAP access method

The following `r3batch.properties` file is an example of a configuration file for the SAP access method with the following characteristics:

- The level of tracing set is DEBUG_MID. This means that not only error messages but also informational messages and warnings are written in the trace file.
- The trace file is created in the `/home/maestro/methods` directory.
- The tracing process creates three trace files, whose maximum size can be 10 MB.

```
r3batch.organization=ABC
r3batch.product=IWS
r3batch.component=R3BATCH
r3batch.trace.tracers.level=DEBUG_MID
r3batch.trace.tracers.listenerNames=r3batch.trace.handlers.traceFile
r3batch.trace.tracers.logging=true
r3batch.trace.handlers.traceFile.fileDir=/home/maestro/methods
r3batch.trace.handlers.traceFile.formatterName=r3batch.trace.formatter
r3batch.trace.handlers.traceFile.maxFileSize=104805100
r3batch.trace.handlers.traceFile.maxFiles=3
```

Part III. Integration with SAP

The following sections give you information about IBM® Workload Scheduler for SAP, the SAP access method and job plug-ins, and how to schedule jobs by using the SAP Solution Manager.

IBM® Workload Scheduler is certified for the following SAP integrations:

- SAP Certified Integration with SAP NetWeaver
- SAP Certified Integration with SAP S/4HANA

Through the integrations, IBM® Workload Scheduler can access SAP BW on HANA Process Chains to handle and manage them externally and can invoke and track InfoPackages.

Chapter 8. Introducing IBM Workload Scheduler for SAP

Improve SAP operations and enable business growth with IBM Workload Scheduler.

Use IBM Workload Scheduler for SAP, to create, schedule, and control SAP jobs using the job scheduling features of IBM Workload Scheduler. IBM Workload Scheduler supported agent workstations help extend the product scheduling capabilities to SAP through the R/3 batch access method . In addition, you can define IBM Workload Scheduler job plug-ins for SAP BusinessObjects BI and SAP PI Channel. With the SAP Solution Manager integration, you can have the IBM Workload Scheduler engine run job scheduling tasks available from the Solution Manager user interface.

IBM Workload Scheduler provides a single and simplified point of planning, control and optimization of end-to-end production services across heterogeneous IT infrastructures. It enables you to control SAP operations from z/OS indifferently.

To understand if a SAP System is compatible with IBM Workload Scheduler, check if your system exposes the following interfaces:

- BC-XBP 6.10 (V2.0) - Background Processing
- BC-XBP 7.00 (V3.0) - Background Processing
- BW-SCH 3.0 – Business Information Warehouse for SAP BW/4HANA
- BW4-SCH - Business Information Warehouse for SAP BW/4HANA

For detailed information about supported SAP interfaces, prerequisite SAP notes, and supported SAP software versions, generate the Data Integration report and select the **Supported Software** tab. In addition, see the dedicated SAP section in the IBM Workload Scheduler Detailed System Requirements.

Features

Table 8: IBM Workload Scheduler for SAP features on page 58 shows the tasks you can perform with IBM Workload Scheduler for SAP either in a distributed or an end-to-end environment, or both.

Table 8. IBM Workload Scheduler for SAP features

Feature	Distributed environment	End-to-end
Using IBM Workload Scheduler standard job dependencies and controls on SAP jobs	✓	✓
Listing jobs, defining jobs, variants, and extended variants using the IBM Workload Scheduler interface	✓	✓
Defining jobs and variants dynamically at run time	✓	✓
Scheduling SAP jobs to run on specified days and times, and in a prescribed order	✓	✓
Scheduling SAP BusinessObjects Business Intelligence (BI) jobs to gain greater control over your SAP BusinessObjects Business Intelligence	✓	✓

Table 8. IBM Workload Scheduler for SAP features (continued)

Feature	Distributed environment	End-to-end
(BI) reports through the IBM Workload Scheduler plug-in for SAP BusinessObjects Business Intelligence (BI).		
Scheduling SAP Process Integration (PI) Channel jobs to control communication channels between the Process Integrator and a backend SAP R/3 system.	✓	✓
Scheduling and monitoring job scheduling tasks available from the SAP Solution Manager user interface.	✓	
Defining the national language support options	✓	✓
Using the SAP Business Warehouse Support functions	✓	✓
Customizing job execution return codes	✓	✓
Using SAP logon groups for load balancing and fault-tolerance	✓	✓
Using Business Component-eXternal Interface Background Processing (XBP 2.0 and later) interface support to:	• Collect intercepted jobs • Track child jobs • Keep all job attributes when you rerun a job • Raise events	• Track child jobs • Keep all job attributes when you rerun a job • Raise events
Using Business Component-eXternal Interface Background Processing (XBP 3.0) interface support to:	• Create criteria profiles to log raised events, reorganize the event history, and intercept and relaunch jobs, according to the criteria you specify. • SAP application log and application return code	• Create criteria profiles to log raised events, reorganize the event history, and intercept and relaunch jobs, according to the criteria you specify. • SAP application log and application return code

Table 8. IBM Workload Scheduler for SAP features (continued)

Feature	Distributed environment	End-to-end
	• Spool list request and display for jobs that have run. • Temporary variants	• Spool list request and display for jobs that have run. • Temporary variants
Assigning an SAP job to a server group, for batch processing	✓	✓
Exporting SAP factory calendars and adding their definitions to the IBM Workload Scheduler database	✓	
Defining internetwork dependencies and event rules for IBM Workload Scheduler based on SAP events	✓	
Defining event rules based on IDoc records	✓	
Defining event rules based on CCMS Monitoring Architecture alerts	✓	
Rerunning a job that submits a process chain from a specific process, from failed processes, or as a new instance	✓	✓
Displaying the details of a job that submits a process chain	✓	✓
Enabling job throttling	✓	✓

Chapter 9. Access method for SAP

The SAP R/3 batch access method enables communication between an external SAP R/3 system and IBM Workload Scheduler and provides a single point of entry for automating the launching of jobs, monitoring the status of jobs and managing exceptions and recovery.

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

IBM Workload Scheduler provides a single and simplified point of planning, control and optimization of end-to-end production services across heterogeneous IT infrastructures. It enables you to control SAP operations from z/OS indifferently.

Scheduling process for the agent workstation hosting the r3batch access method

IBM Workload Scheduler launches jobs in SAP using IBM Workload Scheduler jobs defined to run on a supported agent workstation.

Supported agent workstations include:

- dynamic agents
- extended agents
- IBM Z Workload Scheduler Agents

See Supported agent workstations for more details about these agent workstations.

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

Multiple extended agent workstations can be defined to use the same host, by using multiple options files. Using the SAP extended agent workstation unique identifier as a key, `r3batch` uses the corresponding options file to determine which instance of SAP will run the job. See UNIQUE_ID on page 21 for more information about identifying the extended agent workstation unique identifier. It makes a copy of a template job in SAP and marks the job as "`scheduled`". It then monitors the job through to completion, writing job progress and status information to a job standard list on the host workstation.

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

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

For more detailed information about configuration files on extended agents and dynamic agents, see Configuring the SAP access method on page 77.

Roles and responsibilities

In a typical enterprise, different users contribute to the implementation and operation of the product. Table 9: Roles and responsibilities in IBM Workload Scheduler for SAP on page 62 describes the roles and responsibilities of all users in the process model, showing the tasks they perform.

Table 9. Roles and responsibilities in IBM Workload Scheduler for SAP

User role	User task
IBM Workload Scheduler administrator	• Creating the IBM Workload Scheduler RFC user on page 65 • Creating the authorization profile for the IBM Workload Scheduler user on page 65 • Copying the correction and transport files on page 68 • Importing ABAP/4 function modules into SAP on page 69
IBM Workload Scheduler configurator	• Changing the IBM Workload Scheduler RFC user ID password on page 73 • Migrating from previous versions on page 77 • Print parameter and job class issues on page 75 • Defining the configuration options on page 79 • Connecting to the SAP system on page 98 • Creating SAP Standard R/3 jobs from the Dynamic Workload Console on page 102 • Using the BDC Wait option on page 150 • Implementing job interception on page 151 • Defining user authorizations to manage SAP Business Warehouse InfoPackages and process chains on page 162 • Setting and using job throttling on page 179 • Exporting SAP factory calendars on page 184 • Setting National Language support options on page 219
IBM Workload Scheduler developer	• Editing a standard SAP job on page 109 • Task string to define SAP jobs on page 110 • Displaying details about a standard SAP job on page 120 • Verifying the status of a standard SAP job on page 121 • Deleting a standard SAP job from the SAP database on page 122 • Balancing SAP workload using server groups on page 122 • Defining SAP jobs dynamically on page 128

Table 9. Roles and responsibilities in IBM Workload Scheduler for SAP (continued)

User role	User task
	• Managing SAP Business Warehouse InfoPackages and process chains on page 162 • Defining an IBM Workload Scheduler job that runs an SAP PI Channel job • See the section about prerequisite steps to create SAP BusinessObjects BI in *User's Guide and Reference*.
IBM Workload Scheduler developer	• Defining internetwork dependencies and event rules based on SAP background events on page 187 • Defining event rules based on IDoc records on page 196 • Defining event rules based on CCMS Monitoring Architecture alerts on page 206
IBM Workload Scheduler operator	• Rerunning a standard SAP job on page 126 • Mapping between IBM Workload Scheduler and SAP job states on page 123 • Raising an SAP event on page 125 • Killing an SAP job instance on page 124 • Displaying details about a process chain job on page 168

Configuring user authorization (Security file)

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

For more detailed information about the security file, security file syntax, and how to configure the security file, see "Configuring user authorization (Security file)" in the *Administration Guide*.

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

Table 10. Access keywords for activities with SAP scheduling objects

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

Configuring the SAP environment

You must configure the SAP environment before using the SAP access method.

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

The steps require that you have knowledge of an SAP Basis Administrator.

Overview

About this task

Here is an overview of the customization procedure:

1. Create a new user ID for RFC communications in SAP for IBM Workload Scheduler.
2. Create the authorization profile as described in Creating the authorization profile for the IBM Workload Scheduler user on page 65.
3. Copy the correction and transport files from the IBM Workload Scheduler server to the SAP server.
4. Import the correction and transport files into SAP and verify the installation.

Results

Creating the IBM Workload Scheduler RFC user

About this task

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

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

Creating the authorization profile for the IBM Workload Scheduler user

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

There are two alternative ways to perform this task:

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

Using transaction su02 and manually creating the profile

About this task

Perform the following steps:

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

Object	Description	Authorization
S_ADMI_FCD	System authorizations	S_ADMI_ALL
S_APPL_LOG	Application logs	S_APPL_L_E2E
S_BTCH_ADM	Background processing: Background administrator	S_BTCH_ADM

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

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

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

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

Using transaction PFCG (Profile Generator)

About this task

Perform the following steps:

1. Write a name, for example ZTWS, in **Role Name**.
2. Click **Create Role** and write a description for the role, such as "Role for the TWS user."
3. Save the role.
4. Select **Authorizations**.
5. Click **Change Authorization Data**.

6. In the pop-up, select **Templates**.
7. Manually add the following authorization objects:

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

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

Object	Description
S_ADMI_FCD	System authorizations ◦ System administration function: Full authorization
S_APPL_LOG	Activity: Display ◦ Application log Object name: Full authorization ◦ Application log subobject: Full authorization
S_BTCH_ADM	Background processing: Background administrator ◦ Background administrator ID: Full authorization
S_BTCH_JOB	Background processing: Operations on background jobs ◦ Job operations: Full authorization ◦ Summary of jobs for a group: Full authorization
S_BTCH_NAM	Background processing: Background user name

Object	Description
	◦ Background user name for authorization check: Full authorization
S_PROGRAM	ABAP: Program run checks ◦ User action ABAP/4 program: Full authorization ◦ Authorization group ABAP/4 program: Full authorization
S_RFC	Authorization check for RFC access ◦ Activity: Full authorization ◦ Name of RFC to be protected: Full authorization ◦ Type of RFC object to be protected: Full authorization
S_RZL_ADM	Activity: Full authorization
S_SPO_ACT	Spool: Actions ◦ Authorization field for spool actions: Full authorization ◦ Value for authorization check: Full authorization
S_SPO_DEV	Spool: Device authorizations ◦ Spool - Long device names: Full authorization
S_XMI_LOG	Internal access authorizations for XMI log ◦ Access method for XMI log: Full authorization
S_XMI_PROD	Authorization for external management interfaces (XMI) ◦ XMI logging - Company name: ABC* ◦ XMI logging - Program name: MAESTRO* ◦ Interface ID: Full authorization

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

Copying the correction and transport files

About this task

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

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

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

The names of *control_file* and *data_file* vary from release to release. The files are located in the methods directory: UNIX®: *TWA_DATA_DIR*\methods, Windows™: *TWA_home*\methods, and have the following file names and format:

For SAP releases earlier than 6.10:

- K000*xxx*.TV1 (control file) and R000*xxx*.TV1 (data file)
- K900*xxx*.TV2 (control file) and R900*xxx*.TV2 (data file)

For SAP releases 6.10, or later:

- K9000*xx*.TV1 (control file) and R9000*xx*.TV1 (data file)
- K9007*xx*.TV1 (control file) and R9007*xx*.TV1 (data file)

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

Specifically, for IBM Workload Scheduler version 10.2.4 the following files are used:

For SAP releases earlier than 6.10:

- K000538.TV1 (for standard jobs scheduling)
- R000538.TV1 (for standard jobs scheduling)
- K900294.TV2 (for IDoc monitoring and job throttling)
- R900294.TV2 (for IDoc monitoring and job throttling)

For SAP releases 6.10, or later:

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

Importing ABAP/4 function modules into SAP

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

About this task

This section describes the procedure to generate, activate, and commit new ABAP/4 function modules to your SAP system and several new internal tables. You do not modify any existing SAP system objects. For information about the supported SAP R/3 releases, see the System Requirements Document at Download Documents, System Requirements, Release Notes.

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

- `K9000xx.TV1` (function modules for standard jobs scheduling extensions)
- `K9007xx.TV1` (function modules for IDoc monitoring and job throttling)

where *x* is a digit generated by the SAP system. The methods directory is located in:

On UNIX operating systems

`TWA_DATA_DIR/methods`

On Windows operating systems

`TWA_home\methods`

Before importing the ABAP/4 function modules, review the considerations documented in Migrating from previous versions on page 77.

To import ABAP/4 function modules into SAP:

1. Change to the following directory:

   ```
   cd /usr/sap/trans/bin
   ```

2. Add the transport file to the buffer:

   ```
   tp addtobuffer transport sid
   ```

   where:

   transport

   The transport request file.

   sid

   The SAP system ID.

   For example, if the transport file in the `TWA_home\methods` directory is named `K9000xxx.TV1`, the transport request is tv1K9000*xxx*.

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

   ```
   tp tst transport sid
   ```

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

   If there are errors, check with a person experienced in correction and transport, or try using unconditional modes to do the import.

4. Run the following command to import all the files in the buffer:

   ```
   tp import transport sid
   ```

   This command generates the new ABAP/4 modules and commits them to the SAP database. They automatically become active.

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

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

```
tp import transport sid U126
```

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

   If you apply the standard transport and the IDOC transport, 26 ABAP/4 modules are installed by the import process. For a list of the transport files to be used, refer to Importing ABAP/4 function modules into SAP on page 69. Table 11: ABAP/4 modules installed on page 71 lists the ABAP modules installed.

Table 11. ABAP/4 modules installed

ABAP/4 module	Installed?
ENQUEUE_/IBMTWS/EQ_XAPPL	✓
DEQUEUE_/IBMTWS/EQ_XAPPL	✓
/IBMTWS/UNREGISTER_XAPPL	✓
/IBMTWS/GET_XAPPL_REGISTRATION	✓
/IBMTWS/MODIFY_JOB_CLASS	✓
/IBMTWS/REGISTER_XAPPL	✓
J_1O1_BDC_STATUS	✓
J_1O1_DATE_TIME	✓
J_1O1_IDOC_SELECT	✓
J_1O1_JOB_ADJUST_CLIENT	✓
J_1O1_JOB_FIND	✓
J_1O1_JOB_FINDALL	✓
J_1O1_JOB_HAS_EXTENDED_VARIANT	✓
J_1O1_JOB_LOG	✓
J_1O1_RAISE_EVENT	✓
J_1O1_REPORT_ALL_SELECTIONS	✓
J_1O1_REPORT_GET_TEXTPOOL	✓
J_1O1_VARIANT_COPY	✓
J_1O1_VARIANT_CREATE	✓
J_1O1_VARIANT_DELETE	✓

ABAP/4 module	Installed?
J_1O1_VARIANT_EXISTS	✓
J_1O1_VARIANT_GET_DEFINITION	✓
J_1O1_VARIANT_GET_HELP_VALUES	✓
J_1O1_VARIANT_MAINTAIN_CNT_TBL	✓
J_1O1_VARIANT_MAINTAIN_SEL_TBL	✓
J_1O1_VARIANT_MODIFY	✓

Table 12: ABAP/4 modules contents on page 72 shows the contents of the ABAP modules for the IDoc records and job throttling feature.

Table 12. ABAP/4 modules contents

Object	Description	Used by...
/IBMTWS/	Type = Development Namespace. For IBM Workload Scheduler.	Internal use only
/IBMTWS/EQ_XAPPL	Type = Lock Object. Synchronizes the job throttler instances and job interception collector jobs that are running against the same SAP system.	Job throttling Job interception
/IBMTWS/GET_ XAPPL_REGISTRATION	Type = Function Module. It is used to query for existing external application registration data in table IBMTWS/XAPPL, for example the registration data of a job throttler instance or job interception collector.	Job throttling Job interception
/IBMTWS/MODIFY_JOB_CLASS	Type = Function Module. Modifies the job class of an intercepted job that is controlled by the job throttler. For details, see Step 3. Enabling job class inheritance on page 180.	Job throttling Job interception
/IBMTWS/REGISTER_XAPPL	Type = Function Module. Registers an external application, for example the job throttler.	Job throttling Job interception
/IBMTWS/TWS4APPS	Type = Function group. For IBM Workload Scheduler.	Internal use only
/IBMTWS/UNREGISTER_XAPPL	Type = Function Module. Unregisters an external application, for example the job throttler.	Job throttling Job interception
/IBMTWS/XAPPL	Type = Table. Stores the registration data of external applications. An external application can	Job throttling Job interception

Object	Description	Used by...
	be a job throttler instance or a job interception collector.	
J_1O1_IDOC_SELECT	Type = Function Module. Selects IDoc records from SAP internal tables. For details, see Defining event rules based on IDoc records on page 196.	IDoc event rules
J_1O1_TWS_EDIDC	Type = Data structure in FM interface	Function module J_1O1_IDOC_SELECT
J_1O1_TWS_IDOC_SELECTION	Type = Data structure in FM interface	Function module J_1O1_IDOC_SELECT
J_1O1_TWS_STATE_SELECTION	Type = Data structure in FM interface	Function module J_1O1_IDOC_SELECT

To uninstall the transport you can use the STMS transaction see: Deleting Imported Requests from the Import Queue.

Changing the IBM Workload Scheduler RFC user ID password

About this task

If the password of the IBM Workload Scheduler Remote Function Call (RFC) user ID is modified after the initial installation, the options file used by `r3batch` must be updated to reflect this change.

On UNIX® operating systems, log on as root to the system where IBM Workload Scheduler is installed.

On Windows™ operating systems, log on as an administrator and start a DOS shell on the system where IBM Workload Scheduler is installed.

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

   ```
   enigma new_password
   ```

   where

   new_password

   is the new password for the IBM Workload Scheduler RFC user ID.

   The enigma command prints an encrypted version of the password.

2. Copy the encrypted password into the options file, which is located in the following directory:

   UNIX **On UNIX operating systems**

   TWA_DATA_DIR`/methods`

On Windows operating systems

`TWA_home\methods`

The file can be edited with any text editor.

Results

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

```
{aes}hyb/LQNyVzIf9oA8/xgY+CSqAuAh7+CvTT7HuDpdiu5YUAH0KJEHJtA=
```

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

Securing data communication

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

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

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

Levels of protection

You can apply one of the following levels of protection:

Authentication only

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

Integrity protection

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

Privacy protection

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

Example

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

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

See Defining the local options on page 82 for a description of these options in the local options file.

Print parameter and job class issues

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

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

The following is a list of print parameters supported by BAPI XBP 1.0 for SAP release 4.6*x* and later:

- archiving mode
- authorization
- columns
- delete after output
- lines
- number of copies
- output device
- print immediately
- recipient
- sap cover page
- selection cover page
- spool retention period

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

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

Unicode support

Access method for SAP supports the Unicode standard.

What is Unicode

Unicode was devised to address the problem caused by the profusion of code sets. Since the early days of computer programming hundreds of encodings have been developed, each for small groups of languages and special purposes. As

a result, the interpretation of text, input, sorting, display, and storage depends on the knowledge of all the different types of character sets and their encodings. Programs are written to either handle one single encoding at a time and switch between them, or to convert between external and internal encodings.

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

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

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

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

Unicode support on SAP

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

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

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

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

- Access method for SAP must run on a supported operating system. See the SAP support section in the IBM Workload Scheduler Detailed System Requirements. The product does not support Unicode on the other operating systems where it can be installed.
- The SAP systems that communicate with `r3batch` must be running Unicode-enabled SAP versions.

If these conditions are not met, you cannot use Unicode support and must make sure that `r3batch`, the Dynamic Workload Console, and the target SAP system code page settings are aligned. Use the options related to national language support described in SAP supported code pages on page 220 .

Migrating from previous versions

This version of IBM Workload Scheduler for SAP supports all the SAP versions listed in IBM Workload Scheduler Detailed System Requirements. The IBM Workload Scheduler access method for SAP uses the official SAP RFC interfaces for job scheduling. These are:

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

To avoid conflicts with other vendors, the IBM Workload Scheduler ABAP modules now belong to the IBM Workload Scheduler partner namespace J_1O1_*xxx* and /IBMTWS. After you have completed the imports as described in Importing ABAP/4 function modules into SAP on page 69, the RFC J_1O1_*xxx* function modules and the /IBMTWS function modules are installed on your system.

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

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

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

To upgrade SAP, perform the following steps:

1. Uninstall the IBM® Workload Scheduler ABAP module.
2. Upgrade SAP.
3. Install the IBM® Workload Scheduler ABAP module.

Configuring the SAP access method

This section provides detailed information about the SAP options file creation.

The files for the SAP access method are located in the following path:

UNIX **On UNIX operating systems**

`*TWA_DATA_DIR*/methods`

On Windows operating systems

`TWA_home\methods`

If `r3batch` finds the local configuration file for an extended agent or dynamic agent, it ignores the duplicate information contained in `r3batch.opts`. If instead it does not find a local configuration file then it will use `r3batch.opts` global options file.

To successfully use the SAP access method, you must first install the SAP RFC libraries, as described in the System Requirements Document in the SAP Access Method Requirements section.

Dynamic agents

r3batch.opts

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

DYNAMIC_AGENT_FILE_r3batch.opts

One or more configuration files that are specific to each dynamic agent workstation within a particular installation of a `r3batch` access method. The *`DYNAMIC_AGENT_FILE_r3batch`*`.opts` is the name of the options file, where *DYNAMIC_AGENT* is not necessarily the name of the dynamic agent workstation, because the dynamic agent can have more than one `.opts` file associated. If you do not create a local options file, the global options file is used. Every dynamic agent workstation must have one or more local options file with its own configuration options.

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

Note: If you have a **pool** or **dynamic pool** containing *n* agents, you must create an options file for the dynamic pool and copy it in the *`TWA_home`*`/methods` of each agent of the pool so that all members of the pool have a local options file with the same name. Then you must create another options file for the specific agent in the same directory.

For example, if the SAP access method is installed for **`AGENT1`** and **`AGENT2`** that belong to the dynamic pool ***DYN_POOL***, you need to create the following options files in the path *`TWA_home`*`/methods` on each agent:

AGENT 1

- `FILE_AGENT1_r3batch.opts`
- `FILE_DYN_POOL_r3batch.opts`

AGENT2

- `FILE_AGENT2_r3batch.opts`
- `FILE_DYN_POOL_r3batch.opts`

On dynamic workstations, you can create a new options file or edit an existing options files from a graphical user interface panel available on the General tab of the SAP job definition panels in the Dynamic Workload Console.

Extended agents

r3batch.opts

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

***XA_Unique_ID*_r3batch.opts**

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

Note: *`XA_Unique_ID`* is the unique identifier for the extended agent workstation. See UNIQUE_ID on page 21 for more details about identifying the unique identifier for an extended agent workstation.

For example, to define two extended agent workstations with unique identifiers, `07756YBX76Z6AFX2` and `S4HANAR3BW`, that access two SAP systems, `SAP1` and `SAP2`, with the `r3batch` access method, you must define the following three configuration files:

- Global `r3batch.opts`
- Local file `07756YBX76Z6AFX2_r3batch.opts`
- Local file `S4HANAR3BW_r3batch.opts`

Defining the configuration options

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

Defining the global options

Table 13: r3batch global configuration options on page 80 lists the options that can be specified only in the *global* configuration file `r3batch.opts`.

Table 13. r3batch global configuration options

Option	Description	Default
dep_sem_proj	(Optional) The project ID for the external dependency semaphore used for handling SAP background processing events as external follows dependencies.	`d`
icp_sem_proj	(Optional) The project ID for the job interception semaphore.	`c`
job_sem_proj	(Optional) The project ID for the job semaphore.	`a`
max_jobs_to_release_for_user	Defines the maximum number of jobs released for each user each time the release job is submitted. If this option is less than or equal to 0, the option is ignored and all jobs are released when the release job is submitted. This option is present both in global and local options. If you define it in both configuration files, the local value overrides the global one.	`ON`
primm_enable	(Optional) Enables (`ON`) the SAP print parameter PRIMM (Print Immediately) for all jobs.	`OFF`
prnew_enable	(Optional) Enables (`ON`) the SAP print parameter PRNEW (New Spool Request) for all jobs.	`OFF`
prrel_enable	(Optional) Enables (`ON`) the SAP print parameter PRREL (Immediately delete the spool output after printing) for all jobs.	`OFF`
prsap_enable	(Optional) Enables (`ON`) the SAP print parameter PRSAP (Print SAP Cover Page) for all jobs. The default value is OFF.	`OFF`
prunx_enable	(Optional) Enables (`ON`) the SAP print parameter PRUNX (Print Operating System Cover Page) for all jobs.	`OFF`
release_all_intercepted_jobs_for_request	Releases jobs for each user on a cyclic basis, based on the number of jobs specified in the **max_jobs_to_release_for_user** option. The default value is **ON**, which means that all jobs are submitted for each user: • If the **max_jobs_to_release_for_user** option is less than or equal to **0**, all jobs are released for each user. • If the **max_jobs_to_release_for_user** option is higher than **0**, the specified number of jobs is submitted for each user on a cyclic basis. For example, if **max_jobs_to_release_for_user**=**5**, the first **5** jobs are submitted for each user, then the following 5 jobs for each user, and so on, until all jobs for all users are submitted. If this option is set to **OFF**, it releases for each user only the number of jobs specified in the **max_jobs_to_release_for_user** option. The remaining jobs are submitted only when a new release job is submitted:	`0``0`

Table 13. r3batch global configuration options (continued)

Option	Description	Default
	• If the **max_jobs_to_release_for_user** option is less than or equal to **0**, all jobs are released for each user. • If the **max_jobs_to_release_for_user** option is higher than **0**, only the specified number of jobs is submitted, then no other job is submitted until the new release job. If **max_jobs_to_release_for_user=5**, the first **5** jobs are submitted for each user, then no other job is submitted until the new release job. This option is present both in global and local options. If you define it in both configuration files, the local value overrides the global one.	
var_sem_proj	(Optional) The project ID for the variant semaphore.	b

Note: If the **max_jobs_to_release_for_user** option is higher than 0, only the specified number of jobs is submitted, then no other job is submitted until the new release job. If **max_jobs_to_release_for_user**=5, the first 5 jobs are submitted for each user, then no other job is submitted until the new release job.

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

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

To synchronize on the same semaphore, the communication partners must use the same identifier. There are several ways to choose this identifier. IBM Workload Scheduler for SAP uses two parameters: a path name and a project ID (which is a character value). The path name parameter is the fully qualified path to the options file. The project ID is taken from the options described in Table 13: r3batch global configuration options on page 80. If these options are omitted, IBM Workload Scheduler for SAP uses default values, which work for most installations.

Note:

1. The semaphore options must be edited directly in the global options file using a text editor; you cannot use the options editor to modify these values.
2. If two semaphore options are assigned the same value, all the semaphore values are reset according to the following rule:

   job_sem_proj

   It keeps the value assigned, or its default value.

var_sem_proj

It is reset to the first character that, in the ASCII table, follows the value assigned to `var_sem_proj`.

icp_sem_proj

It is reset to the second character that, in the ASCII table, follows the value assigned to `var_sem_proj`.

dep_sem_proj

It is reset to the third character that, in the ASCII table, follows the value assigned to `var_sem_proj`.

Defining the local options

Table 14: r3batch local configuration options on page 82 lists the options that you can specify only in the *local* configuration files.

Table 14. r3batch local configuration options

Option	Description
bapi_sync_level	(Optional) Specifies the synchronization level between the SAP function modules BAPI_XBP_JOB_COPY and BAPI_XBP_JOB_START_ASAP. Allowed values are: **high** All RFC calls between BAPI_XBP_JOB_START_ASAP and BAPI_XBP_JOB_COPY are synchronized. This is the default. **medium** The RFC calls to BAPI_XBP_JOB_START_ASAP are synchronized. **low** The RFC calls are not synchronized.
blank_libpath	(Optional) Clears (`ON`) the operating system variables LD_LIBRARY_PATH and LIBPATH. The default value is `OFF`.
fn_cache_enabled	(Optional) Enables or disables the file cache on the agent. Can be ON (default value).
fn_cache_purge_interval	(Optional) Specifies the time of validity (in days) of the cached files. If it is left unspecified or set equal to or less than 0, the files are valid indefinitely.
get_job_status_retry	(Optional) Sets the number of times a Remote Function Call must be attempted to retrieve the actual status of an SAP Job. Allowed values are in the range from 1 to 9999. The default value is 5.

Table 14. r3batch local configuration options (continued)

Option	Description
get_job_status_retry_delay	(Optional) Sets the number of seconds between two consecutive calls of a Remote Function Call. Allowed values are in the range from 1 to 9999.
job_duration	(Optional) Enables (`ON`) that the CPU time value in the production plan report that is run from the Dynamic Workload Console is set to the actual duration of the SAP job. Default value is `OFF`. To retrieve the job duration from the SAP system, ensure that the authorization profile contains the following authorization objects: • S_DEVELOP • S_TCODE with parameter SE38 (only for SAP 6.40 and 7.00) For details about the authorization profile, see Creating the authorization profile for the IBM Workload Scheduler user on page 65.
max_jobs_to_release_for_user	Defines the maximum number of jobs released for each user each time the release job is submitted. If this option is less than or equal to 0, the option is ignored and all jobs are released when the release job is submitted. This option is present both in global and local options. If you define it in both configuration files, the local value overrides the global one.
primm_enable	(Optional) Enables (`ON`) the SAP print parameter PRIMM (Print Immediately) for all jobs. The default value is `OFF`.
prnew_enable	(Optional) Enables (`ON`) the SAP print parameter PRNEW (New Spool Request) for all jobs. The default value is `OFF`.
prrel_enable	(Optional) Enables (`ON`) the SAP print parameter PRREL (Print Release) for all jobs. The default value is `OFF`.
prsap_enable	(Optional) Enables (`ON`) the SAP print parameter PRSAP (Print SAP Cover Page) for all jobs. The default value is `OFF`.
prunx_enable	(Optional) Enables (`ON`) the SAP print parameter PRUNX (Print Operating System Cover Page) for all jobs. The default value is `OFF`.
release_all_intercepted_jobs_for_request	Releases jobs for each user on a cyclic basis, based on the number of jobs specified in the **max_jobs_to_release_for_user** option. The default value is **ON**, which means that all jobs are submitted for each user:

Table 14. r3batch local configuration options (continued)

Option	Description
	• If the **max_jobs_to_release_for_user** option is less than or equal to **0**, all jobs are released for each user. • If the **max_jobs_to_release_for_user** option is higher than **0**, the specified number of jobs is submitted for each user on a cyclic basis. For example, if **max_jobs_to_release_for_user**=**5**, the first **5** jobs are submitted for each user, then the following 5 jobs for each user, and so on, until all jobs for all users are submitted. If this option is set to **OFF**, it releases for each user only the number of jobs specified in the **max_jobs_to_release_for_user** option. The remaining jobs are submitted only when a new release job is submitted: • If the **max_jobs_to_release_for_user** option is less than or equal to **0**, all jobs are released for each user. • If the **max_jobs_to_release_for_user** option is higher than **0**, only the specified number of jobs is submitted, then no other job is submitted until the new release job. If **max_jobs_to_release_for_user**=**5**, the first **5** jobs are submitted for each user, then no other job is submitted until the new release job. This option is present both in global and local options. If you define it in both configuration files, the local value overrides the global one.
r3client	(Mandatory) The SAP client number.
r3gateway	(Optional) The host name of the SAP gateway.
r3group	(Optional) The name of the SAP logon group.
r3gwservice	(Optional) The service number of the SAP gateway.
r3host	(Mandatory) The host name of the SAP message server when using logon groups, or the host name of the application server in all other cases. If this server can be reached through one or more SAP gateways, use a string in the format `/H/gateway/H/` for each of them.
r3instance	(Mandatory) The SAP instance number. If **r3group** is set, this option is ignored.
r3password	(Mandatory) The password for the **r3user**. Ensure that you enter the same password when creating this user in the SAP system. It can be a maximum of eight characters and is stored in encrypted format. The value is case sensitive.

Table 14. r3batch local configuration options (continued)

Option	Description
	For information about how to encrypt the password see Encrypting SAP user passwords on page 97.
r3sid	(Mandatory) The SAP system ID.
r3snclib	(Optional) Specifies the path and file name of the SNC library. This option becomes mandatory if **r3sncmode** is activated (1).
r3sncmode	(Optional) Enables (1), or disables (0), secure network communication (SNC) between r3batch and the SAP R3 system. The default setting is (0). Refer to the SAP documentation for more information about using the SAP cryptographic Library for SNC.
r3sncmyname	(Optional) Specifies the name of the user sending the RFC for secure network communication (SNC).
r3sncpartnername	(Optional) Specifies the SNC name of the SAP R3 communication partner (application server). This option becomes mandatory if **r3sncmode** is activated (1).
r3sncqop	(Optional) Specifies the secure network communication (SNC) protection level.
r3user	(Mandatory) The name of the SAP user with which the access method connects to the SAP system. It must have the appropriate privileges for running background jobs. It is sometimes also called the Maestro™ User ID.
report_list_max_limit	(Optional) Sets the maximum number of ABAP reports which can be loaded. The default value is -1, which means no limit.

Defining the common options

Table 15: r3batch common configuration options on page 85 lists additional options that you can specify in either configuration file.

Table 15. r3batch common configuration options

Option	Description	Default
bdc_job_status_failed	(Optional) How IBM Workload Scheduler sets the completion status of a job running BDC sessions, according to a possible BDC processing failure. The allowed values are: ***n*** If at least *n* BDC sessions failed (where *n* is an integer greater than 0), IBM Workload Scheduler sets the job completion status as failed.	`ignore`

Table 15. r3batch common configuration options (continued)

Option	Description	Default
	all If all the BDC sessions failed, IBM Workload Scheduler sets the job completion status as failed. **ignore** When all the BDC sessions complete, regardless of their status, IBM Workload Scheduler sets the job completion status as successful. This is the default. **Note:** This option is ignored if you defined the job by setting the **nobdc** or **nobdcwait** option. For details about these options, see Task string to define SAP jobs on page 110.	
ccms_alert_history	(Optional) Enables (`ON`) or disables (`OFF`) the product to retrieve all the matching CCMS alerts, included those that were generated before the monitoring process started. The default value is `OFF`, meaning that the product retrieves only the CCMS alerts that are generated after the monitoring process started. **Note:** This option takes effect the first time you start the CCMS alert monitoring. If you initially set it to `OFF` and later you want to retrieve the alerts generated before the monitoring process started, stop the monitoring and delete the *`XAname`*`_r3xalmon.cfg` file located in *`TWA_DATA_DIR`*`/methods/r3evmon_cfg` on UNIX® and `TWA_home\methods\r3evmon_cfg` on Windows™. In the options file, set `ccms_alert_history=on` and start the monitoring process again.	`OFF`
commit_dependency	(Optional) Enables (`ON`) or disables (`OFF`) the product to commit internetwork dependencies after processing. If you enable this option, internetwork dependencies are committed immediately by default. If you disable or delete this option, the **-commit** parameter set in the internetwork dependency definition is applied. For details about the **-commit** parameter, see Table 25: Parameters to define an SAP internetwork dependency on page 188.	`OFF`

Table 15. r3batch common configuration options (continued)

Option	Description	Default
enable_appl_rc	(Optional) Enables (`ON`) or disables (`OFF`) the mapping of the application return code to the IBM Workload Scheduler return code. **Note:** This feature does not modify the exit code of the access method. For more details, refer to the **rccondsucc** keyword in the job definition documented in *IBM Workload Scheduler: User's Guide and Reference*.	`OFF`
evmon_interval	(Optional) The polling rate (in seconds) that the `r3evmon` process applies to monitor the list of events.	`60`
ifuser	(Optional) The ID of the user who runs the access method to retrieve job information.	None
idoc_no_history	(Optional) Enables (`ON`) or disables (`OFF`) the product to retrieve only IDoc data that is generated after the monitoring process started. If you specify `OFF`, all matching IDocs are retrieved, including those that were generated before the monitoring process started. When processing this option, `r3evmon` uses the *XAname*`_r3idocmon.cfg` file to retrieve the date and time for the next monitoring loop.	`ON`
idoc_shallow_result	(Optional) Enables (`ON`) or disables (`OFF`) the product to retrieve only the most recent matching IDocs. For example, suppose you set `idoc_shallow_result=ON`. If the status of an IDoc changes several times during the monitoring interval and the same status, matching an event rule condition, occurs more than once in the sequence of statuses, only the most recent matching IDoc is retrieved. If you specify `OFF`, all matching IDocs are retrieved.	`ON`
jobdef	(Optional) If enabled, you can use the Dynamic Workload Console to define jobs, in addition to the command line. Specify `r3batch` to enable the option, and any other value to disable it.	`r3batch`
job_interceptable	(Optional) Enables (`ON`) or disables (`OFF`) the job launched by r3batch to be intercepted by SAP. If enabled, when r3batch launches a job and the SAP job interception feature is enabled, the job can be intercepted if it matches previously defined criteria. If disabled, the job launched by r3batch cannot be intercepted by SAP.	`OFF`

Table 15. r3batch common configuration options (continued)

Option	Description	Default
ljuser	(Optional) The ID of the user who runs the access method to launch jobs (LJ tasks) and manage jobs (MJ tasks).	None
log_r3syslog	(Optional) Enables (`ON`) or disables (`OFF`) the access method to write the latest entries from the SAP syslog to its trace file when an RFC returns with a general error.	`OFF`
long_interval	(Optional) The maximum interval, in seconds, between status checks. It cannot be greater than 3600 seconds. See also *short_interval*.	`3600`
max_n0_counter	(Optional) The maximum value of the N0 counter. If the N0 counter reaches the specified value, it starts again from 0.	`2^15 - 1`
max_name_counter	(Optional) The maximum value of the variant name counter. If the name counter reaches the specified value, it starts again from 0.	`40`
n0_counter_policy	(Optional) The N0 counter policy: **step** The N0 counter is increased once for every step. **job** The N0 counter is increased once for every job.	`job`
name_counter_policy	(Optional) The name counter policy: **step** The name counter is increased once for every step. **job** The name counter is increased once for every job.	`job`
nojobdefs	(Optional) Disables (`1`) or enables (`0`) the definition of new SAP jobs using the Dynamic Workload Console. If this option is set to 1, you must create the job definitions in the SAP job before creating the IBM Workload Scheduler job that is going to schedule them.	`0`
oldcopy	(Optional) Enables (`1`) or disables (`0`) the access method to use the old way of copying jobs, even though the function module BAPI_XBP_JOB_COPY is present on the SAP system.	`0`
pchain_recover	(Optional) The action taken by IBM Workload Scheduler when you rerun a job that submits a process chain. The allowed values are:	`rerun`

Table 15. r3batch common configuration options (continued)

Option	Description	Default
	rerun IBM Workload Scheduler creates another process chain instance and submits it to be run again. **restart** IBM Workload Scheduler restarts the original process chain from the failing processes to the end. For details about rerunning a process chain job, refer to Rerunning a process chain job on page 171.	
pchain_details	(Optional) Enables (`ON`) or disables (`OFF`) the display of details about an SAP process chain that you scheduled as an IBM Workload Scheduler job.	`OFF`
pchainlog_bapi_msg	(Optional) Enables (`ON`) or disables (`OFF`) the product to retrieve additional messages from the BAPI calls from the SAP Business Warehouse process chains and appends them to the stdlist of IBM Workload Scheduler.	`ON`
pchainlog_level	(Optional) Supplements the option **retrieve_pchainlog**. Specifies which level of process chain logs you want to retrieve. Allowed values are: **1** Only the first level of process chain is logged. ***level_number*** Process chains are logged down to the level of chain you indicate here. For example, if you indicate 2 only the first two levels are logged. **all** All process chains are logged.	If you omit this option, and leave **retrieve_pchainlog** set to `ON`, the default is level `1`.
pchainlog_verbosity	(Optional) Supplements the option **retrieve_pchainlog**. Specifies which type of process chain logs you want to retrieve. Allowed values are: **chains_only** Logs only the process chains.	If you omit this option, and leave **retrieve_pchainlog** set to `ON`, the default is `complete`.

Table 15. r3batch common configuration options (continued)

Option	Description	Default
	chains_and_failed_proc In addition to the process chains, logs all failed processes. **complete** Logs all process chains and processes. **Note:** This option affects the entire process chain; verbosity cannot be reduced for individual processes.	
pc_launch_child	(Optional) Enables (`ON`) or disables (`OFF`) the product to launch child jobs that are in scheduled state. **Note:** You can use this option only if you activated the parent-child feature on the SAP system. On the XBP 2.0 or later SAP system, you can activate this feature by using the INITXBP2 ABAP report.	`OFF`
placeholder_abap_step	(Optional) If XBP version 2.0 is used, the name of the ABAP report used as the dummy step in the SAP placeholder job that is created to monitor an SAP event defined as external dependency.	If this option is not specified, either as global or local option, the default `BTCTEST` is used.
qos_disable	(Optional) Enables (`ON`) or disables (`OFF`) the creation of the environment variable QOS_DISABLE on Microsoft™ Windows™ systems that use the Quality of Service (QoS) feature, before `r3batch` opens an RFC connection. Without this option, because of problems in the implementation of the QoS service, the connection between `r3batch` and the SAP RFC library does not work.	`OFF`
r3auditlevel	(Optional) The audit level for the XBP. A number from 0 (low) to 3 (high).	`3`
rcmap	(Optional) Enables (`ON`) or disables (`OFF`) the return code mapping capabilities of Access method for SAP.	`ON`

Table 15. r3batch common configuration options (continued)

Option	Description	Default
retrieve_applinfo	(Optional) Enables (`ON`) or disables (`OFF`) the retrieval and appending of the SAP application log to the stdlist of IBM Workload Scheduler.	`OFF`
retrieve_ipaklog	(Optional) Enables (`ON`) or disables (`OFF`) the retrieval and appending of the SAP BW InfoPackage logs to the `stdlist` of IBM Workload Scheduler. **Note:** The retrieval and appending of SAP BW InfoPackage job logs to the `stdlist` might be time-consuming for jobs that produce large logs.	`ON`
retrieve_joblog	(Optional) Enables (`ON`) or disables (`OFF`) the retrieval and appending of the SAP job logs to the `stdlist` of IBM Workload Scheduler. **Note:** 1. The retrieval and appending of job logs to the `stdlist` might be time-consuming for jobs that produce large logs. 2. If you disable the retrieval of the job logs, you also disable the return code mapping function for the log entries. 3. This option does not affect the `BDC Wait` feature.	`ON`
retrieve_pchainlog	(Optional) Enables (`ON`) or disables (`OFF`) the retrieval and appending of the SAP BW process chain logs to the `stdlist` of IBM Workload Scheduler. **Note:** 1. The retrieval and appending of SAP BW process chain logs to the `stdlist` might be time-consuming for jobs that produce large logs. 2. If you disable the retrieval of the SAP BW process chain logs, you also disable the return code mapping function for the log entries.	`ON`

Table 15. r3batch common configuration options (continued)

Option	Description	Default
	3. This option on its own retrieves the log of only the first level of a process chain. To retrieve more complete logs, use this option with the **pchainlog_level** and **pchainlog_verbosity** options.	
retrieve_spoollist	(Optional) Enables (`ON`) or disables (`OFF`) the retrieval and appending of the SAP job spool lists to the `stdlist` of IBM Workload Scheduler. **Note:** 1. The retrieval and appending of SAP job spool lists to the `stdlist` might be time-consuming for jobs that produce large spool lists. 2. If you disable the retrieval of the SAP job spool lists, you also disable the return code mapping function for the spool list entries.	`ON`
retry	(Optional) The retry count for SAP function module calls. Specify an integer greater than 0.	`5`
rfc_interval	(Optional) The polling rate (in milliseconds) with which r3batch listens for results of RFC requests. The rate cannot exceed 1,000 milliseconds. Consider that the lower the value of the rfc_interval option, the higher the frequency with which RFC request results are collected and, as a consequence, CPU consumption on the r3batch system is high.	`10`
rfc_open_delay	(Optional) The maximum number of seconds to wait between two consecutive calls before opening an RFC connection.	`1800`
rfc_open_retry	(Optional) The retry count for opening an RFC connection to the SAP system. Specify an integer greater than 0 to limit the number of retries, or `-1` for an unlimited number of retries.	`5`
rfc_timeout	(Optional) The time (in seconds) that `r3batch` waits before canceling a non-responding RFC communication. Allowed values are in the range from 0 to 9999; 0 means no timeout.	`600`

Table 15. r3batch common configuration options (continued)

Option	Description	Default
short_interval	(Optional) The minimum interval, in seconds, between status checks. It cannot be less than 2 seconds. Setting this option to low values makes the notification of status changes faster, but increases the load on the hosting machine. See also *long_interval*.	`10`
throttling_enable_ job_class_inheritance	(Optional) Enables (`ON`) or disables (`OFF`) the inheritance of priority class. `ON` means that the intercepted job inherits the priority class of its progenitor job, if it is higher than its own class; otherwise it keeps its own class. `OFF` means that the intercepted job keeps its own class, regardless of its progenitor's class. **Note:** By setting this option, the parent-child feature is automatically enabled on the SAP system.	`ON`
throttling_enable_ job_interception	(Optional) Enables (`ON`) the job interception feature at job throttler startup, or keeps the current setting (`OFF`). `ON` means that when the job throttler starts, it enables the job interception feature on the SAP system. When the job throttler is stopped, the job interception feature is also automatically restored to the setting that was previously configured on the SAP system. `OFF` means that the job interception feature is kept as it is currently set in the SAP system.	`ON`
throttling_job_ interception_ version	Specifies the BC-XBP interface version to be used when the job throttler starts. Valid values are: • 2 • 3 The default BC-XBP interface version that is used is `2` (version 2.0).	`2`
throttling_interval	(Optional) The interval (in seconds) between each job throttling run.	`5`
throttling_max_ connections	(Optional) The maximum number of connections (connection pool size) that the job throttler can open to communicate with the SAP system. The minimum value is 3.	`5`
throttling_release_all_ on_exit	(Optional) Enables (`ON`) or disables (`OFF`) the release of all intercepted jobs.	`ON`

Table 15. r3batch common configuration options (continued)

Option	Description	Default
	`ON` means that when the job throttler is stopped, it releases all the intercepted jobs. `OFF` means that when the job throttler is stopped, it does not release the intercepted jobs therefore the jobs remain intercepted, in scheduled state.	
throttling_send_ ccms_data	(Optional) Enables (`ON`) or disables (`OFF`) the sending of data from job throttling to the SAP CCMS Monitoring Architecture. `ON` means that the job throttler sends its status data to CCMS continuously. `OFF` means that the job throttler does not send its status to CCMS.	`OFF`
throttling_send_ ccms_rate	(Optional) Rate (in number of runs) at which the job throttler sends its status data to the SAP CCMS monitoring architecture. The minimum value is `1`, meaning that the job throttler sends the data at every run.	`1`
twsmeth_cp	(Optional) The code page that `r3batch` uses to write its output. This option must be consistent with **twsmeth_lang**. It can be any of the existing TIS codepages.	The code page used by the IBM Workload Scheduler workstation that hosts the `r3batch` access method.
twsmeth_lang	(Optional) The language used to report messages. This option must be consistent with **twsmeth_cp**.	The language of the locale of the workstation that hosts the `r3batch` access method.
twsxa_cp	(Optional) The encoding used by `r3batch` to establish RFC connections with SAP systems. Use this option if `r3batch` is not Unicode-enabled. Possible values are: • 1100 • 1103 • 8000 • 8300 • 8400	`1100`
twsxa_lang	(Optional) The language used to log in to SAP systems. Specify one of the following (DE, EN, and JA can be set from the Option Editor. The other languages can be set using any text editor):	`EN`

Table 15. r3batch common configuration options (continued)

Option	Description	Default
	DE German **EN** English **ES** Spanish **FR** French **IT** Italian **JA** Japanese **KO** Korean **pt_BR** Brazilian Portuguese **zh_CN** Simplified Chinese **zh_TW** Traditional Chinese 	
use_fips	(Optional) Enables (ON) or disables (OFF) the FIPS mode of operation for IBM Workload Scheduler.	`OFF`
utf8cmdline	(Optional) Enables (`1`) or disables (`0`) the encoding of extended parameters in UTF-8 format. The default value is `0`.	**Note:** If you have both global and

Table 15. r3batch common configuration options (continued)

Option	Description	Default
		 local options files and you want to change the default value for **utf8cmdline**, modify the local options file because this overrides the global options.
variant_delay	(Optional) The time, in seconds, that r3batch allows the SAP system to clean up the structures used for communication between r3batch and the SAP system. This option is valid when you launch a job that uses extended variants and requires a copy of a job template. Use this option only when you want to reduce r3batch response time, because it increases the load on the hosting machine. Higher values of **variant_delay** increase the response time and decrease the load. Allowed values are in the range from 0 to 3600.	`10`
variant_selection_ screen	(Optional) Specifies the functional interface used to read report selection screens. Specify one of the following: **Custom** To communicate with the SAP system using the IBM® Workload Scheduler custom function module. **SAP** To communicate with the SAP system using the XBP 3.0 function module.	Custom
xbpversion	(Optional) The XBP version used on the target SAP system. Specify an integer value. This value overwrites the XBP version automatically determined during RFC logon.	The XBP version determined by r3batch during RFC

Table 15. r3batch common configuration options (continued)

Option	Description	Default
	Note: For details about XBP 3.0 and SAP NetWeaver 2004s with SP9, refer to the SAP Note 977462.	logon from the SAP system.

SAP option file example

Below is an example of an options file for SAP. It can help you determine your specific site requirements, although your options file might be different.

```
r3client=100
r3host=/H/tiraix64.lab.rome.abc.com
r3instance=00
r3password={aes}hyb/LQNyVzIf9oA8/xgY+CSqAuAh7+CvTT7HuDpdiu5YUAH0KJEHJtA=
r3sid=GS7
r3user=twstest
long_interval=120
r3auditlevel=3
short_interval=10
twsxa_lang=EN
```

Encrypting SAP user passwords

When you add your entries in the options file from the Dynamic Workload Console, the password value is automatically encrypted before it is written in the file. If you modify the file with a text editor, run the **enigma** program to encrypt the password before writing it in the file, as follows:

```
enigma your_password
```

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

Configuration options usage

The format of the configuration file is the following:

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

with no blanks before the option, after the value, or before or after the equals (=) character.

You can put all the common information, such as the `LJuser`, `IFuser`, `JobDef`, and `LogFileName` options in `r3batch.opts`, while you can put tailored data for the target SAP system of the extended agent or dynamic agent (for example, `SAP1`) in a local configuration file (for example, `XA1_r3batch.opts`).

You can put a local option in the global configuration file if you want to give the same option to all the `r3batch` instances. For example, if the SAP user name is the same in all your SAP systems, you can place the `r3user` option in the global file without duplicating that information in all the local configuration files.

A global option, such as `job_sem_proj`, only has effect in the global configuration file. If you put global options in a local file they have no effect.

`r3batch` reads the global configuration file first and then the local file. Every option (except the global options) contained in the local configuration file will override those in the global file. For example, if both the global and the local configuration files contain the `r3user` option, `r3batch` uses the one in the local file.

There are six mandatory options that `r3batch` requires:

- `r3client`
- `r3host`
- `r3instance`
- `r3password`
- `r3sid`
- `r3user`

You can put them all in the local configuration file or you can spread them between the global and the local files. For example, you could put `r3user` and `r3password` in the global configuration file and `r3sid`, `r3instance`, `r3client`, and `r3host` in the local one.

The `r3user` option is both local and mandatory. It must be put either in the global configuration file or the local configuration file.

Note: These configuration files are not created during the installation process.

Connecting to the SAP system

The Access method for SAP uses the SAP remote connection call (RFC) library to connect to the SAP system. The connection address for an SAP system is denoted as a `connection string`.

To successfully use the SAP R/3 access method, you must first install the SAP RFC libraries, as described in the System Requirements Document in the SAP R/3 Access Method Requirements section.

Connecting to a specific application server

To connect to a specific application server, you enter strings which, according to the complexity of the networks, might be more or less complex and contain passwords to secure the routers.

In its basic form, a connection string consists of the host name (or IP name) of an SAP application server; for example:

```
/H/hemlock.romlab.rome.abc.com
```

This type of connection string works only in very simple network environments, where all application servers can be reached directly through TCP/IP. Usually, modern companies use more complex network topologies, with a number of small subnetworks, which cannot communicate directly through TCP/IP. To support this type of network, the SAP RFC library supports SAP routers, which are placed at the boundaries of the subnetworks and act as proxies. For this type of network, the connection string is a composite of basic connection strings for each SAP router, followed by the basic connection string for the target SAP system; for example:

```
/H/litespeed/H/amsaix33/H/hemlock.romlab.rome.abc.com
```

Moreover, you can secure the SAP routers with passwords, to prevent unauthorized access. In this case, the basic connection string for the SAP router is followed by `/P/` and the password of the router.

Note: The SAP RFC library limits the length of the connection string to a maximum of 128 characters. This is a real limitation in complex network environments. As a workaround, it is recommended to use simple host names, without the domain name whenever possible. Alternatively, you can use the IP address, but this is not recommended, because it is difficult to maintain.

IBM Workload Scheduler for SAP supports both types of connection strings, basic and composite, where:

r3host

The connection string.

r3instance

The SAP instance number.

r3sid

The SAP system ID.

For example:

```
r3host=/H/litespeed/H/amsaix33/H/hemlock.romlab.rome.abc.com
r3instance=00
r3sid=TV1
```

Connecting to a logon group

About this task

In large SAP installations, the application servers are usually configured in logon groups for load balancing and fault-tolerance purposes. Load balancing is done by a dedicated server, called the `message server`. The message server automatically assigns users to the application server with the least workload of the logon group it controls.

Ensure that the file services (on UNIX: `/etc/services` on Windows: `C:\Windows\system32\drivers\etc\services`) contain an entry for the message server port of the SAP system to which `r3batch` connects. The entry has the following format:

```
sapms<SID> 36<system_number>/tcp
```

where *`SID`* is the SAP system ID, and *`system_number`* is the SAP system number.

Set the following options to configure `r3batch` to connect to a logon group:

r3host

The hostname of the message server.

r3group

The name of the logon group.

r3sid

The SAP system ID.

For example:

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

Configuring SAP event monitoring

This section provides detailed information about how to configure your system to monitor SAP events:

- Prerequisite to defining event rules based on SAP events on page 100
- Monitoring SAP events on page 101

Prerequisite to defining event rules based on SAP events

About this task

To be able to define event rules based on one or more SAP events, stop the IBM Workload Scheduler WebSphere Application Server Liberty Base and copy the following file (located on the system where you installed IBM Workload Scheduler:

On UNIX operating systems

TWA_DATA_DIR/methods/SAPPlugin/SapMonitorPlugIn.jar

On Windows operating systems

TWA_home\methods\SAPPlugin\SapMonitorPlugIn.jar

to the following directory of the master domain manager and of its backup nodes:

On UNIX operating systems

TWA_DATA_DIR/eventPlugIn

On Windows operating systems

TWA_home\eventPlugIn

For the changes to take effect, stop and restart the IBM Workload Scheduler WebSphere Application Server Liberty Base. If the master domain manager is connected to the Dynamic Workload Console, stop and restart also the Dynamic Workload Console Application Server.

Monitoring SAP events

Whenever you define an event rule based on an SAP event in your IBM Workload Scheduler plan, that event is monitored by IBM Workload Scheduler. Monitoring SAP events is allowed only if you use XPB version 3.0, or later.

IBM Workload Scheduler monitors two types of SAP event:

Events defined by the SAP system

The events that are triggered automatically by system changes, for example when a new operation mode is activated. This type of event cannot be modified by the user.

Events defined by the user

The events that are triggered by ABAP or external processes, for example when a process triggers an SAP event to signal that external data has arrived and must be read by the SAP system. For details about how to trigger events by external processes, refer to Raising an SAP event on page 125.

If you modify the `r3batch` option files, to make the changes effective you must stop and restart the extended agent monitoring processes with the following command. For UNIX only, this command must be entered by the owner of the IBM Workload Scheduler installation:

Command syntax

```
r3evman { start | stop }
```

Where:

start | stop

The action to perform:

start

Starts monitoring SAP events.

stop

Stops monitoring SAP events.

Defining SAP jobs

You must define some jobs to be able to run jobs on an SAP workstation from IBM Workload Scheduler.

To define and manage jobs on an SAP workstation from IBM Workload Scheduler, you must define the following:

Jobs in SAP that you want to run under IBM Workload Scheduler control

You can define these jobs using standard SAP tools or using the Dynamic Workload Console.

Jobs in IBM Workload Scheduler that correspond to the jobs in SAP

The IBM Workload Scheduler job definitions are used in scheduling and defining dependencies, but the SAP jobs are actually run.

You can define SAP job definitions from the Dynamic Workload Console and then have IBM Workload Scheduler launch the jobs in SAP R/3 using jobs defined on the following workstations that support the r3batch access method:

- An IBM Workload Scheduler extended agent workstation. A workstation that is hosted by a fault-tolerant agent or master workstation.
- A dynamic agent workstation.
- A dynamic pool.
- A z-centric workstation.

You can manage your SAP environment from both:

- An IBM Workload Scheduler distributed environment
- An IBM Z Workload Scheduler environment.

The SAP job definitions can reference the following types of SAP jobs:

- Standard R/3
- Business Warehouse Process Chains
- Business Warehouse InfoPackages

For information about Business Warehouse Process Chains and Business Warehouse InfoPackages, see Using Business Information Warehouse on page 161.

Creating SAP Standard R/3 jobs from the Dynamic Workload Console

How to create and manage an SAP job that is associated to an IBM Workload Scheduler job that manages it.

About this task

You can easily create and manage Standard R/3 jobs on a remote SAP system entirely from the Dynamic Workload Console, and then continue to manage the remote SAP job from IBM Workload Scheduler.

The IBM Workload Scheduler job definition, available for both distributed and z/OS environments, maps to the newly created job on the SAP system. The SAP job can run on extended agent workstations, dynamic agent workstations, pools, dynamic pools, and workstations depending on the type of job definition you choose to create.

Note: Using this procedure to create a new IBM Z Workload Scheduler Agent SAP Standard R/3 job, you cannot manage variants. To manage variants, use the SAP graphical user interface or use the **List Jobs on SAP** entry from the navigation tree of the Dynamic Workload Console.

To create a new SAP Standard R/3 job on a remote SAP system that maps to an IBM Workload Scheduler job definition, you have to associate your SAP Standard R/3 jobs to IBM Workload Scheduler jobs and you can do as follows:

- Creating an SAP job on a remote SAP system: Creating an SAP job from the Dynamic Workload Console on page 103
- Creating an IBM Workload Scheduler job and associating it to an SAP job: Create an IBM Workload Scheduler job and associate it to an SAP job on page 103

When performing operations that require a connection to a remote SAP system, you must configure the SAP connection data. The connection is made through an IBM Workload Scheduler workstation with the r3batch access method installed. Each workstation can have one or more options files that can be used to customize the behavior of the r3batch access method, except for extended agent workstations, where only one options file can be defined and therefore a selection is not required. For information about setting the SAP connection data, see Setting the SAP data connection on page 105.

Creating an SAP job from the Dynamic Workload Console

How to create an SAP job definition on a remote SAP system from the Dynamic Workload Console.

About this task

You can also create and save SAP Standard R/3 jobs directly on the remote SAP system from IBM Workload Scheduler, as you would from the SAP graphical user interface. To create Standard R/3 jobs on the SAP system from the Dynamic Workload Console, perform the following steps:

1. From the **Design** menu, select **Manage Jobs on SAP**.
2. In the Job Filter Criteria, select **Standard Job** and specify the workstation name. This parameter is mandatory because it identifies the remote SAP system.
3. Specify the workstation where the SAP job runs. This is the workstation with the r3batch access method that communicates with the remote SAP system.
4. If the workstation is not an extended agent workstation, you must also specify the options file to be used.
5. Click **Display** to view a list of the Standard R/3 jobs for the specified workstation.
6. Click **New** to create a new Standard R/3 job and enter the required information.
7. Click **Save** to create the job on the SAP system.

What to do next

After creating the new SAP job on the SAP from the Dynamic Workload Console, you must reference it in an IBM Workload Scheduler SAP Standard R/3 job if you want to manage the job from within IBM Workload Scheduler as explained in Create an IBM Workload Scheduler job and associate it to an SAP job on page 103.

Create an IBM Workload Scheduler job and associate it to an SAP job

Create an IBM Workload Scheduler job definition and map it to a new or existing SAP job to manage it.

About this task

To create a new IBM Workload Scheduler job and then associate it to a new SAP job, follow these steps:

1. From the **Design** menu, click Workload Designer page.
2. Select an engine. The Workload Designer window is displayed.

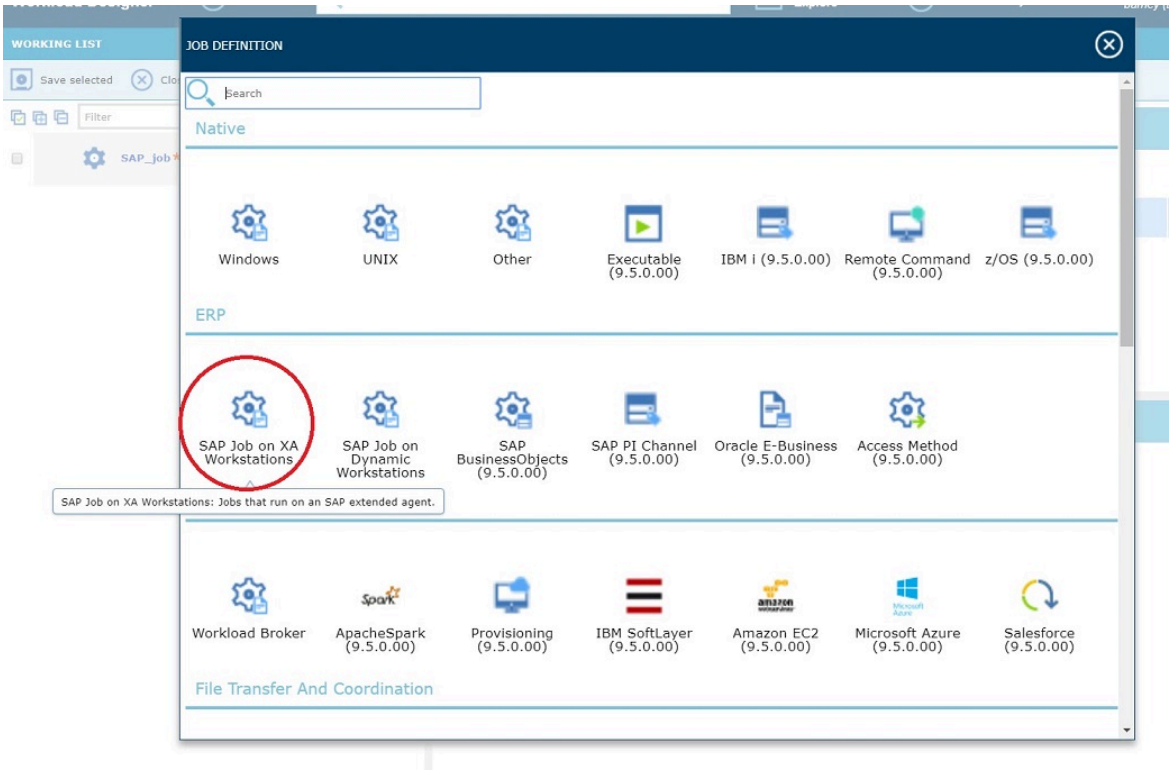
3. Click **Create New**, select **Job definition** and in the **ERP** category choose:
   - Distributed

     SAP Job on Dynamic Workstations

     For distributed systems only. This job definition can run on dynamic agent workstations, dynamic pools, and IBM Z Workload Scheduler Agent workstations.

     SAP Job on XA Workstations

     This job definition can run on extended agent workstations, which are workstations hosted by fault-tolerant agents or master workstations.



   - z/OS **SAP**

     SAP

     For z/OS systems only. This job definition references an existing job on the SAP system and can run on dynamic agent workstations, dynamic pools, and IBM Z Workload Scheduler Agent.

4. In the Properties pane, specify the properties for the SAP job definition you are creating using the tabs available. The tabs for each type of SAP job definition are similar, but there are some differences depending on the type of engine you selected and the type of workstation on which the job runs. For more detailed information about the UI elements on each tab, see the Dynamic Workload Console online help.
5. In the **Task** section, specify the IBM Workload Scheduler job that you want to associate to the SAP job.
6. Click **Save** to create the SAP job definition in the IBM Workload Scheduler database.

Setting the SAP data connection

You can configure a default connection to be used when performing actions that access the remote SAP system.

About this task

There are several operations you can perform which require connection details to establish a link to a remote SAP system. The connection is made through an IBM Workload Scheduler workstation with the r3batch access method installed used to communicate with the SAP system. Each workstation can have one or more options files that can be used to customize the behavior of the r3batch access method, except for extended agent workstations, where only one options file can be defined and therefore a selection is not required.

For example, you can use **Workload Designer** to create IBM Workload Scheduler job definitions that reference remote SAP jobs, or you can create a SAP job on a remote SAP system. You can also search for SAP jobs on the remote system from the **Working List** and **Quick Open** panes.

To configure a default SAP data connection to be used when creating items in the Workload Designer that require a SAP connection, perform the following steps:

1. From the **Design** menu, select **Manage SAP Criteria Profile**.
2. Select an engine.
3. In **Workstation**, enter the name of the workstation that communicates with the SAP system. If you do not know the name of the workstation, click the lookup icon and specify a search criteria to find the workstation.
4. In **Options file**, enter the file name of the options file or click the lookup icon to search for options files that reside on the specified workstation and select one.
5. Click **Go** to establish the connection.

Results

A default SAP connection is now configured. It will be used each time an item that requires access to a SAP system is defined.

Managing SAP variants using the Dynamic Workload Console

Managing variants using the Dynamic Workload Console.

About this task

This section describes how to manage variants using the Dynamic Workload Console:

1. Click **Design > SAP > Manage Jobs on SAP** from the portfolio.
2. Specify an engine connection.
3. In **Workstation name**, type the name of the workstation where the SAP job runs. This is the workstation with the r3batch access method that communicates with the remote SAP system. If you do not know the name of the workstation, click (...) browse to enter your filter criteria and click **Search**. If you enter a string representing part of the workstation name, it must be followed by the asterisk (*) wildcard character. Both the question mark (?) and asterisk (*) are supported as wildcards. You can also simply use the asterisk wildcard character (*) to display all workstations. Optionally, specify any of the other search criteria available and click **Search**.

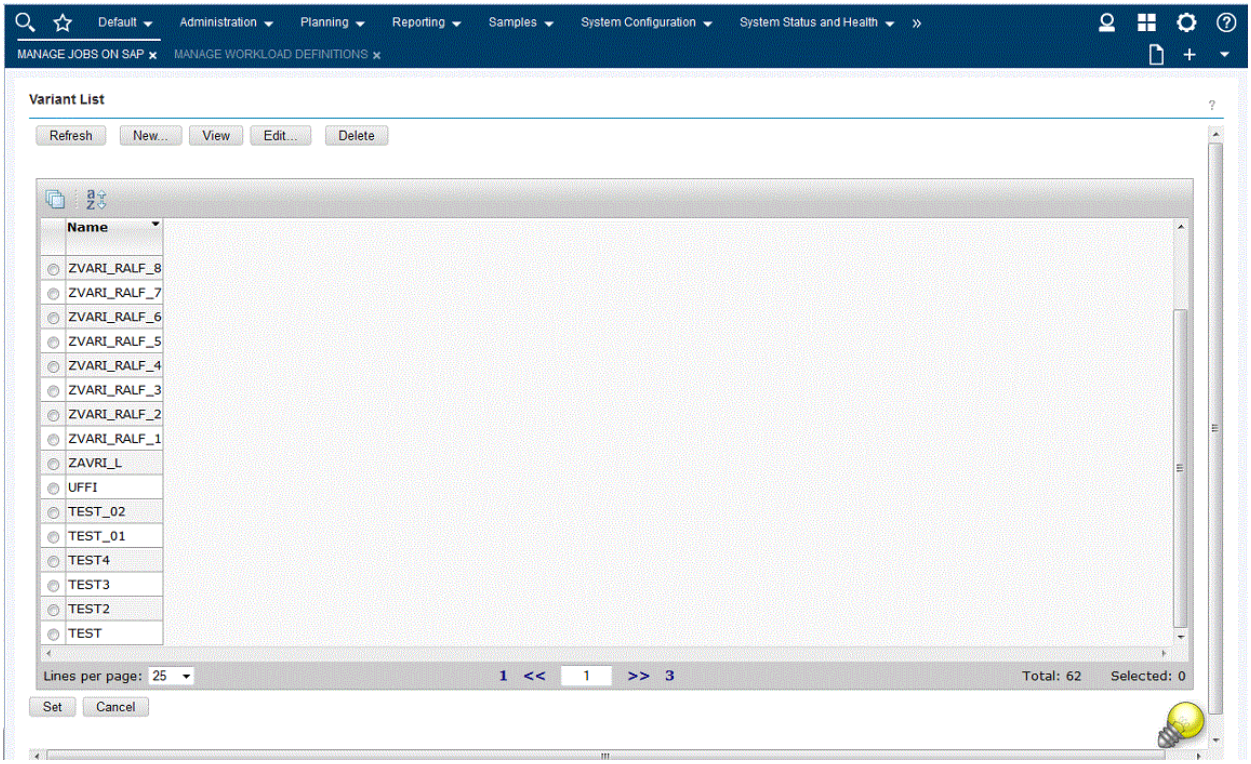
Result

From the results displayed, select the workstation and click **OK**.

4. In **Options file**, specify an options file that resides on the specified workstation. Each workstation can have one or more options files that can be used to customize the behavior of the r3batch access method, except for extended agent workstations, where only one options file can exist and therefore does not need to be specified. For the workstation specified, enter the file name of the options file or click the browse (...) button to search for options files that reside on the specified workstation and select one.
5. Click **Display**. The list of available jobs on the remote SAP system for the specified engine is displayed.
6. A list of SAP jobs on the remote SAP system are displayed.
7. Select a SAP job from the list and click **Edit**.
8. On the **SAP Steps** page, select a program of type ABAP from the list and click **Edit**. The properties for the ABAP program are displayed.
9. In the **Variant** field, click the ellipsis (**...**) icon to display the Variant List panel. This panel lists all the variants associated with the ABAP specified in the **Name** field.

   Result

   Figure 4. The Variant List panel



10. From this panel, you can take the following actions:

    Refresh

    To refresh the content of the variant list with the information contained in the SAP database.

    New

    To create a new variant as described in Creating or editing a variant on page 107.

View

To display information on an existing variant.

Edit

To modify information on an existing variant as described in Creating or editing a variant on page 107.

Delete

To delete a variant.

Set

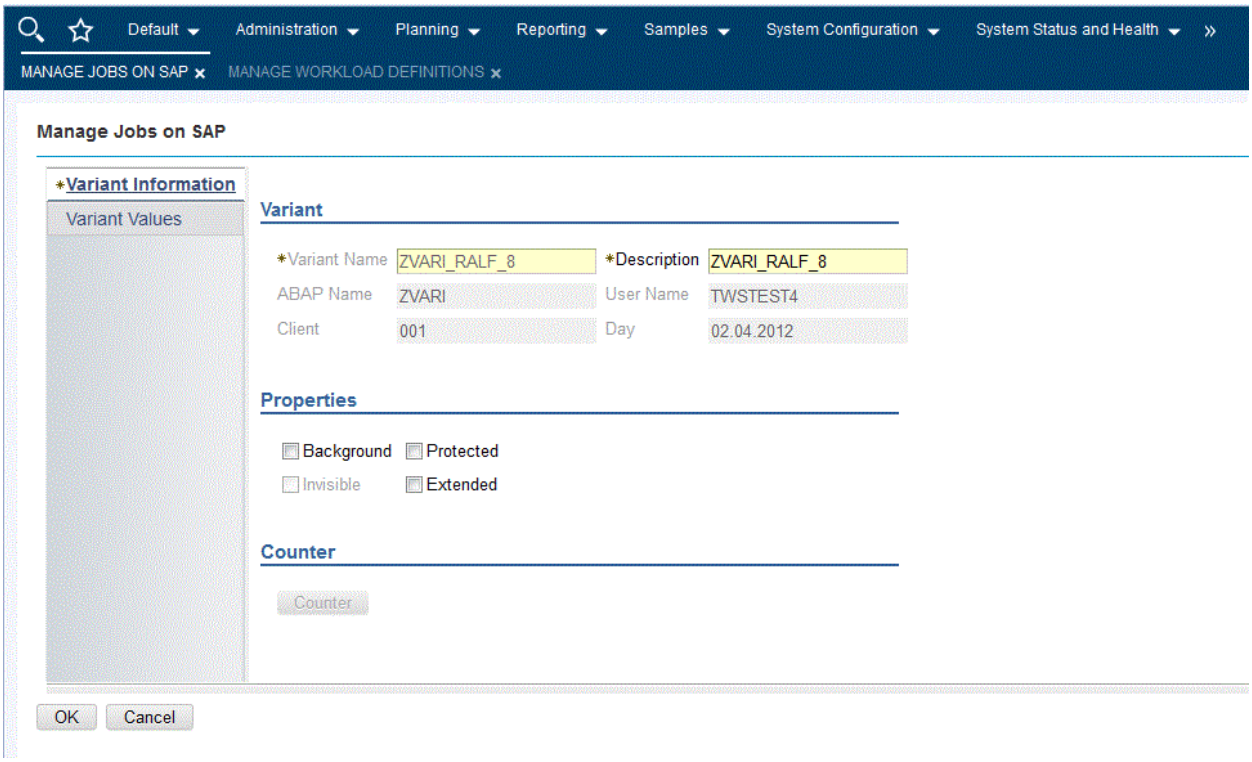
To associate the value chosen from the list to the ABAP.

Creating or editing a variant

About this task

You can create or edit a variant from the **Variant List** panel. To display the **Variant List** panel, see Managing SAP variants using the Dynamic Workload Console on page 105.

1. In the Variant List panel, click **New** or **Edit**. The Variant Information page is displayed by default. If you are editing an existing variant, the fields and selections are not empty.
   Figure 5. The Variant Information page of the Variant List panel



2. The panel consists of the following pages:

◦ Variant Information:

a. Enter or modify the variant name and description.

b. Optionally, check a **Properties** box:

Background

The variant can only be used in background processing.

Protected

The variant is protected against being changed by other users.

Invisible

The variant will not be displayed in the F4 value list on the SAP GUI. Not available for the BC-XBP 3.0 interface.

Extended

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

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

Table 16. Placeholders and counters for extended variants

Symbol	Meaning	Syntax
$S	Timestamp	YYYYMMDDHHMM
$D	Day of the month	DD
$_D	Date	YYYYMMDD
$M	Month	MM
$Y	Year	YY
$_Y	Year	YYYY
$H	Hour	HH
$T	Minute	MM
$_T	Time	HHMMSS
$N*x*	Counters	10 counters: $N0 - $N9 ($N = $N0)

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

- Variant Values:

  In the Variant Values page, the fields and values are dynamically built through `r3batch` depending on the characteristics of the variant or step and are identical to the ones in the equivalent SAP panel.

Editing a standard SAP job

Before you begin

You can edit SAP Standard R/3 jobs in two different ways in IBM Workload Scheduler.

- The Dynamic Workload Console contains the **Manage Jobs on SAP** entry in the portfolio for creating and editing SAP Standard R/3 jobs on remote SAP systems.
- From the Workload Designer you can create and edit remote SAP jobs. See Creating SAP Standard R/3 jobs from the Dynamic Workload Console on page 102.

About this task

To edit a SAP standard R/3 job, follow these steps:

1. Click **Design > SAP > Manage Jobs on SAP**.
2. Select the name of the engine connection from which you want to work with SAP jobs.
3. Leave the default setting in the **SAP Job Type** section to **Standard R/3 Job**.
4. In **Workstation name**, type the name of the workstation where the SAP job runs. This is the workstation with the r3batch access method that communicates with the remote SAP system. If you do not know the name of the workstation, click (...) browse to enter your filter criteria and click **Search**. If you enter a string representing part of the workstation name, it must be followed by the asterisk (*) wildcard character. Both the question mark (?) and asterisk (*) are supported as wildcards. You can also simply use the asterisk wildcard character (*) to display all workstations. Optionally, specify any of the other search criteria available and click **Search**.
   Result
   From the results displayed, select the workstation and click **OK**.
5. In **Options file**, specify an options file that resides on the specified workstation. Each workstation can have one or more options files that can be used to customize the behavior of the r3batch access method, except for extended agent workstations, where only one options file can exist and therefore does not need to be specified. For the workstation specified, enter the file name of the options file or click the browse (...) button to search for options files that reside on the specified workstation and select one.
6. Click **Display**. The list of available jobs on the remote SAP system for the specified engine is displayed.

7. Select the job you want to modify in the list and click **Edit**. The **List Jobs on SAP** panel is displayed.
8. Edit the properties on the **R/3 Job Definition** and **R/3 Steps** pages as appropriate. Refer to the contextual online help available for more detailed information about the UI elements available on each page.

   Note:
   - On the **R/3 Job Definition** page, when you modify the **Job Class**, **Target Host**, or **Server Group** and click **OK**, the **Job ID** is maintained and remains synchronized with the one associated to the current job. Instead, when you modify the **Job Name** and click **OK**, the **Job ID** is automatically replaced with the one associated to the new job name.
   - On the **R/3 Steps** page, for each step you modify, the new step information is saved in the SAP database. For each step you add or delete, the **Job ID** is maintained and remains synchronized with the one associated to the modified step.

9. Click **OK** to save your changes.

Task string to define SAP jobs

This section describes the task string parameters that define and control the running of SAP jobs. You can specify them in the following places when you define their associated IBM Workload Scheduler jobs:

- In the **SAP Command Line** section of the Task page of the Submit Ad Hoc Jobs action from the Dynamic Workload Console.
- In the **SAP Command Line** field of the More Options page of the SAP job definition, if you use the Dynamic Workload Console and selected a **SAP** job definition.
- As arguments of the `scriptname` keyword in the job definition statement, if you use the IBM Workload Scheduler command line.
- As arguments of the `JOBCMD` keyword in the `JOBREC` statement in the `SCRIPTLIB` of IBM Z Workload Scheduler, if you are scheduling in an end-to-end environment. The following is an example of a `JOBREC` statement:

  ```
  JOBREC
    JOBCMD('/-job job_name -user user_name -i job_ID -c class_value')
    JOBUSR(TWS_user_name)
  ```

  where:

  class_value

  The priority with which the job runs in the SAP system. For details, see Table 17: Task string parameters for SAP jobs on page 111.

  job_ID

  The unique SAP job ID. For details, see Table 17: Task string parameters for SAP jobs on page 111.

  job_name

  The name of the SAP job to run. For details, see Table 17: Task string parameters for SAP jobs on page 111.

user_name

The SAP user who owns the target job. For details, see Table 17: Task string parameters for SAP jobs on page 111.

TWS_user_name

The IBM Z Workload Scheduler user who runs the `r3batch` access method from the end-to-end scheduling environment.

The string syntax is the following:

Job definition syntax

```
-jobjob_name [{ -i | -id} job_ID ][ -useruser_name ][{ -host | -ts} host_name ][ -sgserver_group ]
[ -clientsource_client ][ -exec_clienttarget_client ][ -rfc_clientrfc_logon_client ][ -cclass_value ][
 -bdc_job_status_failed bdc_processing ][{ -nobdc | -nobdcwait }][ -bapi_sync_level{high | medium | low }][
 -sstarting_step_number ][ -s Step_number attribute_name [ =attribute_value ]][ -vStep_number variant_name ]
[ -vtxtStep_number variant_description ][ -vparStep_number name=variant_value ][ -vselStep_number name={ i |
e } #operation#lowest [ #highest ]][ -vtemp Step_number ][ -recipientR/3_login_name ][ -rectyperecipient_type ]
[ -flag {reccp | recbl}][ -flag recex][ -flag recnf][ -flag{im | immed}][ -flag{enable_applinfo |
disable_applinfo}][ -flag{enable_appl_rc | disable_appl_rc}][ -flag{enable_joblog | disable_joblog}][
 -flag{enable_job_interceptable | disable_job_interceptable}][ -flag{enable_spoollist | disable_spoollist}][
 -flagpc_launch][{ -debug}][ -tracelvl{1 | 2 | 3}][{ -rfctrace}]
```

Table 17: Task string parameters for SAP jobs on page 111 describes the parameters for the task string to define SAP jobs.

Note:

1. You can specify both `-i` or `-id` and `-user` in the same job definition, but the user name is ignored.
2. When you specify the job ID, both `-client` and `-exec_client` are ignored because the ID is unique for the entire SAP system.
3. Typically, the `-debug` and `-trace` options are for debugging the extended agent and should not be used in standard production.

Table 17. Task string parameters for SAP jobs

Section	Parameters	Description	GUI Support
JOB	-job *job_name*	The name of the job to run. This parameter is mandatory.	✓

Table 17. Task string parameters for SAP jobs

(continued)

Section	Parameters	Description	GUI Support
	-i *job_ID* -id *job_ID*	The unique SAP job ID. Specify this parameter if you are submitting a job that refers to a predefined job template stored in the SAP database for which you want to change a parameter.	✓
	-user *user_name*	The SAP user who owns the target job. Use this parameter when the target SAP system has only one job with the specified name for the specified user. This parameter has no effect if a job ID is specified in the job definition.	✓
	-host *host_name* -ts *host_name*	The name of the SAP workstation where the job is to be run. *host_name* has the format *hostname_SAPsystemname_SAPsystemnumber*. For example, the name of a host might be amss80a0_gs7_90 These parameters are mutually exclusive with -sg.	✓
	-sg *server_group*	The name of the SAP server group where the job is to be run. Use this parameter to run the job on an application server that belongs to the group. The server group must exist on the SAP system, otherwise an error code is returned and the job is not launched. This parameter is case-sensitive and can be up to 20 characters. It is mutually exclusive with -host and -ts.	✓
JOB	-client *source_client*	The number that identifies the SAP client where the job definition is to be found, regardless of the client number defined by the `r3client` keyword in the options file. This parameter has no effect if a job ID is specified in the job definition.	
	-exec_client *target_client*	The number that identifies the SAP client where the job is to be run, regardless of the client number defined by the r3client keyword in the options file. This parameter has no effect if a job ID is specified in the job definition.	

Table 17. Task string parameters for SAP jobs

(continued)

Section	Parameters	Description	GUI Support
	-rfc_client *rfc_logon_client*	The number that identifies the SAP client to be used for RFC logon. This value overwrites the value specified by the r3client keyword in the corresponding r3batch options file.	
	-c *class_value*	The priority with which the job runs in the SAP system. Possible values are: **A** High priority **B** Medium priority **C** Low priority. This is the default value.	✓
	-bdc_job_status_failed *bdc_processing*	How IBM Workload Scheduler sets the completion status of a job running BDC sessions, according to a possible BDC processing failure. The allowed values are: ***n*** If at least *n* BDC sessions failed (where *n* is an integer greater than 0), IBM Workload Scheduler sets the job completion status as failed. **all** If all the BDC sessions failed, IBM Workload Scheduler sets the job completion status as failed. **ignore** When all the BDC sessions complete, regardless of their status, IBM Workload Scheduler sets the job completion status as successful. This is the default. If **-nobdc** or **-nobdcwait** is set, this option is ignored.	✓

Table 17. Task string parameters for SAP jobs

(continued)

Section	Parameters	Description	GUI Support
	-nobdc -nobdcwait	Disables the BDC Wait option (enabled by default) to have the job considered as completed even if not all its BDC sessions have ended.	✓
	-bapi_sync_level	Specifies the synchronization level between the SAP function modules BAPI_XBP_JOB_COPY and BAPI_XBP_JOB_START_ASAP. Allowed values are: **high** All RFC calls between BAPI_XBP_JOB_START_ASAP and BAPI_XBP_JOB_COPY are synchronized. This is the default. **medium** The RFC calls to BAPI_XBP_JOB_START_ASAP are synchronized. **low** The RFC calls are not synchronized.	
STEP	-s *starting_step_number*	The number of the starting step.	✓
	-s*Step_number attribute_name=attribute_value*	The step number and its attributes, where: ***step_number*** The number of the step being defined. Each step is identified by a sequential number (1, 2, 3, ...*n*) using the *step number* ***attribute_name*** The name of the attribute. ***attribute_value*** The value of the attribute. It is optional for some attributes. Attributes can be defined in any order, but cannot be repeated for the same step. Attribute validation is performed before the job is created in the SAP system. If	✓

Table 17. Task string parameters for SAP jobs

(continued)

Section	Parameters	Description	GUI Support

		the validation fails, the IBM Workload Scheduler job goes into the ABEND state. For a detailed description of each attribute and its value, see Defining attributes for ABAP steps on page 139 and Defining attributes for external programs and external commands steps on page 142.
		
		For example, the following step (step 8) is an ABAP module running the report MYPGM and has two attributes, only one of which has a value.
		`-s8 type=A -s8 program=MYPGM`
		`-s8 pr_cover="My title" -s8 pr_immed`	

VARIANT	-v*step_number* name	The variant name for the specified step number.	✓
	-vtxt*step_number variant_description*	The textual description of the variant, in the IBM Workload Scheduler logon language (customizable with the `TWSXA_LANG` option of `r3batch`). The maximum length is 30 characters.	✓
	-vpar*step_number* name=*value*	For ABAP modules only. The value for a variant parameter for the specified step number. This parameter is mandatory when creating a new variant. See Defining attributes for ABAP steps on page 139 for a complete list of the supported attributes for ABAP steps.	✓
	-vsel*step_number* name=*sign#operation#lowest*[*#highest*]	For ABAP modules only. The value for a variant selection option for the specified step number. **sign** Sign of the operation. Possible values are: **I** Include **E** Exclude **operation** Possible values are:	✓

Table 17. Task string parameters for SAP jobs

(continued)

Section	Parameters	Description	GUI Support
		EQ Equals **NE** Not equal to **BT** Between **NB** Not between **LT** Less than **LE** Less than or equal to **GT** Greater than **GE** Greater than or equal to **CP** Contains pattern **NP** Does not contain pattern **lowest** Low value of the selection. You can use up to 45 characters. **highest** High value of the selection. You can use up to 45 characters. This attribute is optional. For a complete list of the supported attributes for ABAP steps, seeDefining attributes for ABAP steps on page 139.	

Table 17. Task string parameters for SAP jobs

(continued)

Section	Parameters	Description	GUI Support
	-vtemp*step_number*	For ABAP modules only. Specifies to assign a temporary variant to the specified step number. Temporary variants are created ad-hoc by the SAP system and assigned to the job instance when it is run. The lifecycle of the temporary variant is determined by the SAP system. If the job is deleted by SAP, then the temporary variant is deleted. See Examples: Dynamically defining and updating SAP jobs on page 144 to refer to examples that demonstrate the behavior of temporary variants.	
SPOOL	-recipient *name*	The login name of an SAP user.	
	-flag {reccp|recbl}	Specifies how the spool list is sent to the recipient. Possible values are: **reccp** The spool list is sent as a copy. **recbl** The spool list is sent as a blind copy.	
	-flag recex	Specifies that the spool list is sent as an express message to the recipient.	
	-flag recnf	Specifies that the recipient is not allowed to forward the spool list.	
	-rectype type	Specifies the recipient type. Possible values are: ' ' SAP user (default value) **'B'** SAP user **'C'** Shared distribution list **'D'** X.500 address	

Table 17. Task string parameters for SAP jobs

(continued)

Section	Parameters	Description	GUI Support
		'G' Organization object/ID **'H'** Organization unit **'I'** SAP object **'L'** Telex number **'O'** SAPoffice user **'P'** Private distribution list **'R'** SAP user in another SAP system **'U'** Internet address **'1'** Other recipient type	
FLAGS	-flag im -flag immed	Specifies to launch job immediately, meaning that if there are no spare work processes, the job fails.	✓
	-flag enable_applinfo -flag disable_applinfo	Enables or disables the retrieval and appending of the SAP application log to the stdlist of IBM Workload Scheduler.	✓
	-flag enable_appl_rc -flag disable_appl_rc	Enables or disables the mapping of the SAP application return code to the IBM Workload Scheduler return code. The SAP application return code is mapped only if **-flag enable_applinfo** is set and the application log contains the application return code.	

Table 17. Task string parameters for SAP jobs

(continued)

Section	Parameters	Description	GUI Support
	-flag enable_joblog -flag disable_joblog	Enables or disables retrieval of the joblog.	✓
	-flag enable_job_interceptable -flag disable_job_interceptable	Enables or disables the job launched by r3batch to be intercepted by SAP. If enabled, when r3batch launches a job and the SAP job interception feature is enabled, the job can be intercepted if it matches previously defined criteria. If disabled, the job launched by r3batch cannot be intercepted by SAP. This setting overwrites the setting in the common options file.	✓
	-flag enable_spoollist -flag disable_spoollist	Enables or disables retrieval of the spool lists of the job.	✓
-flag pc_launch	Specifies to launch child jobs that are in scheduled state. **ON** The product launches child jobs that are in scheduled state. **OFF** The product does not launch child jobs that are in scheduled state. This is the default value. **Note:** You can use this option only if you activated the parent-child feature on the SAP system. On the XBP 2.0 (or later)SAP system you can activate this feature using the INITXBP2 ABAP report		
TRACING	-debug	Enables maximum trace level.	✓

Table 17. Task string parameters for SAP jobs

(continued)

Section	Parameters	Description	GUI Support
	-tracelvl 1|2|3 -rfctrace -trace	Specifies the trace setting for the job. Possible values are: **1** Only error messages are written in the trace file. This is the default. **2** Informational messages and warnings are also written in the trace file. **3** A most verbose debug output is written in the trace file. For detailed information, refer to Configuring the tracing utility on page 54. Enables RFC trace.	✓

The following is an example for an SAP job named `BVTTEST` with ID `03102401` and user `myuser`:

```
-job BVTTEST -i 03102401 -user myuser -debug
```

Managing SAP jobs

This section describes how to manage SAP jobs.

Displaying details about a standard SAP job

About this task

Perform the following steps to display details for standard jobs on specific workstations.

For information about how to display details about a job that submits an SAP process chain, refer to Displaying details about a process chain job on page 168.

1. Click **Design > SAP > Manage Jobs on SAP**.
2. In **Engine name**, select the name of the IBM Workload Scheduler engine connection from which you want to view SAP job details.

3. In **Workstation name**, type the name of the workstation where the SAP job runs. This is the workstation with the r3batch access method that communicates with the remote SAP system. If you do not know the name of the workstation, click (...) browse to enter your filter criteria and click **Search**. If you enter a string representing part of the workstation name, it must be followed by the asterisk (*) wildcard character. Both the question mark (?) and asterisk (*) are supported as wildcards. You can also simply use the asterisk wildcard character (*) to display all workstations. Optionally, specify any of the other search criteria available and click **Search**.
   Result
   From the results displayed, select the workstation and click **OK**.
4. In **Options file**, specify an options file that resides on the specified workstation. Each workstation can have one or more options files that can be used to customize the behavior of the r3batch access method, except for extended agent workstations, where only one options file can exist and therefore does not need to be specified. For the workstation specified, enter the file name of the options file or click the browse (...) button to search for options files that reside on the specified workstation and select one.
5. Click **Display**. The list of available jobs for the specified engine is displayed.
6. Select the job for which you want to display the details and click **Details**. The List Jobs on SAP panel is displayed containing job and time information.
7. When you have finished viewing the details for the job, click **OK** to return to the list of SAP jobs on the workstation specified.

Verifying the status of a standard SAP job

About this task

To verify the status of a standard SAP job, perform the following steps:

1. Click **Design > SAP > Manage Jobs on SAP**.
2. In **Engine name**, select the name of the IBM Workload Scheduler engine connection from which you want to verify the status of an SAP job.
3. In **Workstation name**, type the name of the workstation where the SAP job runs. This is the workstation with the r3batch access method that communicates with the remote SAP system. If you do not know the name of the workstation, click (...) browse to enter your filter criteria and click **Search**. If you enter a string representing part of the workstation name, it must be followed by the asterisk (*) wildcard character. Both the question mark (?) and asterisk (*) are supported as wildcards. You can also simply use the asterisk wildcard character (*) to display all workstations. Optionally, specify any of the other search criteria available and click **Search**.
   Result
   From the results displayed, select the workstation and click **OK**.
4. In **Options file**, specify an options file that resides on the specified workstation. Each workstation can have one or more options files that can be used to customize the behavior of the r3batch access method, except for extended agent workstations, where only one options file can exist and therefore does not need to be specified. For the workstation specified, enter the file name of the options file or click the browse (...) button to search for options files that reside on the specified workstation and select one.
5. Click **Display**. The list of available jobs for the specified engine is displayed.

6. Select the job for which you want to verify the status and click **Status**. The current status for the SAP job is displayed, as well as the database name where the job is installed.
7. When you have finished verifying the status for the job, click **OK** to return to the list of SAP jobs on the workstation specified.

Deleting a standard SAP job from the SAP database

About this task

To delete a standard SAP job from the SAP database, perform the following steps:

1. Click **Design > SAP > Manage Jobs on SAP**.
2. In **Engine name**, select the name of the IBM Workload Scheduler engine connection from which you want to delete the SAP job.
3. In **Workstation name**, type the name of the workstation where the SAP job runs. This is the workstation with the r3batch access method that communicates with the remote SAP system. If you do not know the name of the workstation, click (...) browse to enter your filter criteria and click **Search**. If you enter a string representing part of the workstation name, it must be followed by the asterisk (*) wildcard character. Both the question mark (?) and asterisk (*) are supported as wildcards. You can also simply use the asterisk wildcard character (*) to display all workstations. Optionally, specify any of the other search criteria available and click **Search**.
   Result
   From the results displayed, select the workstation and click **OK**.
4. In **Options file**, specify an options file that resides on the specified workstation. Each workstation can have one or more options files that can be used to customize the behavior of the r3batch access method, except for extended agent workstations, where only one options file can exist and therefore does not need to be specified. For the workstation specified, enter the file name of the options file or click the browse (...) button to search for options files that reside on the specified workstation and select one.
5. Click **Display**. The list of available jobs for the specified engine is displayed.
6. Select the job or jobs you want to delete and click **Delete**. A confirmation message prompts you to confirm the delete action.
7. When the delete action is complete, click **OK** to return to the list of SAP jobs on the workstation specified.

Balancing SAP workload using server groups

SAP jobs run on application servers that host work processes of type *batch*. Critical batch jobs are run in specific time frames, on specific application servers. With SAP Basis version 6.10 and later, application servers can be assigned to server groups. With IBM Workload Scheduler you can assign a server group to a job. In this way, when a job is launched, the SAP system runs it on an application server that belongs to that group, balancing the workload among the various application servers in the group.

If the application servers defined in a group are modified in the SAP system, the job defined as belonging to that server group is not affected and does not need to be modified. The batch execution targets are reorganized in the SAP system without having to change job definitions in IBM Workload Scheduler.

This function is supported with the following versions of SAP:

- SAP Basis 6.10, with Service Pack 40
- SAP Basis 6.20, with Service Pack 41
- SAP Basis 6.40, with Service Pack 04
- SAP Basis 7.00, and later

Mapping between IBM Workload Scheduler and SAP job states

When an SAP job is launched by IBM Workload Scheduler, you can monitor its progress. The status transitions in IBM Workload Scheduler (internal status) and the corresponding SAP status are listed in Table 18: Status transitions in IBM Workload Scheduler (internal status) and the corresponding SAP R/3 status on page 123.

Table 18. Status transitions in IBM Workload Scheduler (internal status) and the corresponding SAP R/3 status

IBM Workload Scheduler Job State	SAP Job State
INTRO	Not Available
WAIT	Ready, Release
EXEC	Active
SUCC	Finished
ABEND	Canceled

The INTRO state indicates that IBM Workload Scheduler is in the process of introducing the job, but in SAP, the job has not yet entered the ready state. Because it takes some time to get a job queued and into the ready column, the INTRO state might last a few minutes if the SAP system is particularly busy.

Even if a job is finished in SAP, IBM Workload Scheduler keeps it in the EXEC state if its BDC sessions are not complete and you have not selected the **Disable BDC Wait** option. For details about this option, see Using the BDC Wait option on page 150.

Managing spools

Browse spool lists on request without having to download the entire spool which can occupy significant space on the file system.

Spool lists can be very large so rather than download them as part of a job run, you can request to browse the spool list, chunks at a time, even if you have disabled the option, retrieve_spoollist, to append the spool list to the IBM Workload Scheduler joblog.

From the Dynamic Workload Console, you can list the spool data available for SAP Standard R/3 jobs that have run. Each spool is identified by the following information:

- The spool number.
- The related step number.

- The name of the spool request.
- The title of the spool request.
- The total number of pages for the spool information.
- The user who executed the SAP job related to the spool.
- The date the spool was created based on the Coordinated Universal Time (UTC) time standard.
- The client for which the spool was created.

Browsing spool data

You can list the spool data available for SAP Standard R/3 jobs that have run and browse the contents of the spool.

About this task

To browse spool data for a specific job that has run:

1. From the **Monitoring and Reporting** menu, click **Orchestration Monitor**.
2. In the **Monitor Workload** input fields enter the engine name, the plan, and any filtering data that helps you filter the selection of jobs (you can also select **Edit** for a guided selection of filtering criteria) and select **Run**.
3. In the output table select an SAP Standard R/3 job and click **More Actions > Show Spool List**. The list of spool data available for the selected job is displayed.
4. Select a spool and click **Spool**.

Results

By default, the first ten pages of the spool are made available. You can change this default by editing the number of pages specified in **Pages for screen**. Use the page functions to jump to a specific page number, jump to the last page of the spool, jump to the first page of the spool, or move forward or back through the number of pages indicated by **Pages for screen**.

Killing an SAP job instance

About this task

This section describes how to kill an IBM Workload Scheduler job that submits either a standard SAP job or an SAP process chain.

To kill an SAP job instance, do the following:

The IBM Workload Scheduler job status is set to ABEND. The SAP job or process chain is set to canceled in the SAP system.

Note: If you kill a process chain job, the SAP system stops as soon as the process that is currently running completes.

1. Use the **Monitor Workload** query of the Dynamic Workload Console to display a list of defined job instances containing the job you want to kill. From the **Monitoring and Reporting** menu, click **Orchestration Monitor**.
2. In the **Monitor Workload** input fields enter the engine name, the plan, and any filtering data that helps you filter the selection of jobs (you can also select **Edit** for a guided selection of filtering criteria) and select **Run**.
3. The **Monitor Jobs** panel is displayed. Select the job instance you want to kill and click **More Actions > Kill**.

Raising an SAP event

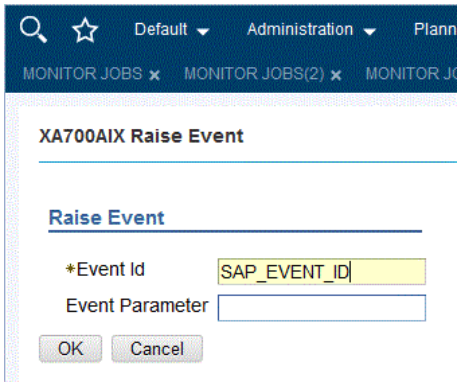
About this task

You can raise events on XBP 2.0 (or later) SAP jobs in the IBM Workload Scheduler database in one of the following ways:

Using the Monitor Workload in the Dynamic Workload Console

Perform the following steps:

1. On the SAP system, create a job that has as start condition a SAP event. When you create this job, its status is released.
2. Check that this job was not intercepted by the interception function.
3. Log in to the Dynamic Workload Console.
4. From the **Monitoring and Reporting** menu, click **Orchestration Monitor**.
5. In the Monitor Workload window select the engine, enter **Workstation** in the Object Type field, and select the plan to display the list of workstations you want to monitor. Click **Run**.

   A list of workstations is displayed.
6. Select a workstation that has been defined to connect to a remote SAP system.
7. From the toolbar, select **More Actions > Raise Event**. The Raise Event panel opens.
   Figure 7. The Raise Event panel
   

8. The panel consists of the following:

   Event ID

   The identifier of the event that is to be raised.

   Event Parameter

   The parameter of the event that is to be raised.
9. Click **OK**. The event is raised.

Creating a job that launches a Windows™ or UNIX™ command that raises an event

Perform the following steps:

1. From the Design menu, click Workload Designer page.
2. Specify an engine name, either distributed or z/OS. The Workload Designer window opens. Job types and characteristics vary depending on whether you select a distributed or a z/OS engine.
3. Click **Create new** and select **Job definition**.
4. Select the **Native** category and then either **Windows** or **UNIX** job.
5. Use the **General** section to provide general information about the new job definition.
6. Use the **Task** section to provide task information for the job.
7. In the **Task** section, select **Command** and in the command string type the following command that raises the event:

   ```
   <data_dir>/methods/r3event -c workstation_name -u user_name
   -e SAP_event_ID -p parameter
   ```

   where:

   workstation_name

   The name of the workstation where the SAP job is defined.

   user_name

   The name of the SAP user with which the access method connects to the SAP system. This is the name specified in the r3user option.

   SAP_event_ID

   The identifier of the event.

   parameter

   The parameter defined for the event.
8. Save the job definition.

What to do next

See Defining conditions and criteria on page 146 for information about how to define criteria that manages which raised events to log.

Rerunning a standard SAP job

You can rerun a standard SAP job from the start, or from a specific numeric step of the SAP instruction.

About this task

To rerun a standard SAP job, you can use one of the following user interfaces:

conman

For details, refer to the *IBM Workload Scheduler: User's Guide and Reference*.

Dynamic Workload Console

Dynamic Workload Console

For details about how to rerun a job that submits an SAP process chain, refer to Rerunning a process chain job on page 171.

For an SAP extended agent, a `step` is the numeric step of the SAP instruction from which a job can be restarted. Before you rerun an SAP job with IBM Workload Scheduler, you have the option of providing a step name for the job. This affects `r3batch` in the following ways:

- If you use a step name that is up to 9 digits (or 8 digits preceded by a character) in length, this name is used as the starting step number for the rerunning job.
- If you use any different format, the name is ignored and the job is rerun starting from the first step.

For example, to rerun a job from the third step, you can use: `A03`, `3`, `00003`, or `H3`.

z/OS In z/OS environments, you need to set the status of the job to **Ready** before you can rerun the job.

Note: By default, if you specify a job step to rerun, the new job is assigned the name of the step you indicated. To keep the original job name, set the IBM Workload Scheduler global option **enRetainNameOnRerunFrom** to `yes`. This option works only when used with the following arguments: `rr jobselect;from=[wkstat#]job`. For details about these arguments, see *IBM Workload Scheduler: User's Guide and Reference*, *Managing objects in the plan - conman*, *Conman commands*, *rerun*. For details about this option, see *IBM Workload Scheduler: Planning and Installation*.

When `r3batch` reruns a job from its first step, either because you specified it as the starting step or because no starting step was specified, it uses the `new copy` feature, if applicable. If the starting step is greater than one, `r3batch` uses the `old copy` to rerun the job. For a description about the difference between the new and old copy of a rerunning job, refer to Old copy and new copy of a rerunning job on page 128.

To rerun a SAP Standard R/3 job from the Dynamic Workload Console, perform the following steps:

1. In the Dynamic Workload Console select **Monitoring & Reporting > Workload Monitoring > Monitor Workload**.
2. In the Monitor Workload input fields select **Job** as the Object Type, the engine name, the plan, and any filtering data that helps you filter the selection of jobs (you can also select **Edit** for a guided selection of filtering criteria) and select **Run**.
3. A list of jobs is displayed. Select an SAP Standard R/3 job.
4. Rerun the job.

   Distributed **Distributed environment**

   a. Click **Rerun...**. The General properties for the rerun operation are displayed.
   b. Optionally, you can choose to not rerun the same job but instead, substitute the selected SAP job with a different job definition and run it. Type the job definition name in the **From Job Definition** field, or use the browse button to search for it and select it.
   c. Optionally, type the workstation name of the workstation on which you want to rerun the job in the **Workstation Name** field.
   d. Optionally, in **Step**, enter a specific numeric step of the SAP instruction from which you want to rerun the job rather than rerunning the whole job.

e. Optionally, specify the start and finish time for the job.
f. Click **Rerun**.

The job reruns immediately or at the specified start time.

z/OS environment

In a z/OS environment, an alias for the job name is not required so the job reruns with the same name. The list of jobs always reports the latest action performed on the job.

a. Before you can rerun a job, you must change the status of the job to **Ready**. Select a job and click **Set Status**.
b. In **Change Status**, select **Ready**.
c. Click **OK** to return to the list of jobs.

The job reruns immediately and the internal status reports **Started**.

Old copy and new copy of a rerunning job

When the access method for SAP launches a job, it makes a copy of a template job and runs it.

The `new copy` feature is available for SAP versions 3.1i, and later. It copies an entire job, preserving steps, job class, and all print and archive parameters. It is performed by using a new SAP function module that is part of the SAP Support Package as stated in the SAP Notes 399449 and 430087.

The `old copy` feature, instead, is based on standard SAP function modules, and creates a new SAP job and adds the steps with a loop that starts from the step name or number you specified. Be aware that, unless you have XBP 2.0 or later:

- The `old copy` does not preserve all the print and archive parameters.
- The job class of the copy is always set to class `C`.

Refer to Print parameter and job class issues on page 75 to learn how to resolve the problem of lost job class and print and archive parameters.

SAP Note 758829 is required to ensure correct operation of the `new copy` and `old copy` features. See also Table 42: Miscellaneous troubleshooting items on page 221.

Defining SAP jobs dynamically

This section describes how to create and submit SAP jobs dynamically without creating or referencing predefined job templates.

When you launch a job created as described in Creating SAP Standard R/3 jobs from the Dynamic Workload Console on page 102 and Task string to define SAP jobs on page 110, IBM Workload Scheduler makes a copy of the predefined job (also known as a template job) and runs the copy. If you want to run the job on several SAP systems, you must manually create the template job on each system.

To create and submit SAP jobs dynamically, without creating or referencing predefined job templates, submit:

- In the SAP system, a job that does not reference an existing template in the SAP R/3 database.
- A job that references a predefined job template stored in the SAP R/3 database for which you want to change a parameter.

To take full advantage of this feature, make sure that you have XBP version 2.0 or later installed, because earlier versions of XBP do not support the full set of print and archive parameters, or provide a way to set the job class or the spool list recipient.

Task string to define SAP jobs dynamically

This section describes the task string that controls the running of SAP jobs. You can build an entire job definition by using the six main sections concerning SAP job parameters. These sections are grouped in Table 19: Task string parameters for SAP jobs (dynamic definition) on page 130 and are related to the:

- Job
- Job steps
- Variants associated with the steps (for ABAP modules only)
- Spool list recipients associated with the job
- Flags associated with the job
- Tracing specifications for the job

You can specify them in the following places when you define their associated IBM Workload Scheduler jobs:

- In the **SAP Command Line** section of the Task page of the Submit Ad Hoc Jobs action from the Dynamic Workload Console.
- In the **SAP Command Line** field of the More Options page of the SAP job definition, if you use the Dynamic Workload Console and selected a **SAP** job definition.
- As arguments of the `scriptname` keyword in the job definition statement, if you use the IBM Workload Scheduler command line.
- As arguments of the `JOBCMD` keyword in the `JOBREC` statement in the `SCRIPTLIB` of IBM Z Workload Scheduler, if you are scheduling in an end-to-end environment. The following is an example of a `JOBREC` statement:

```
JOBREC
  JOBCMD('/-job job_name -user user_name -i job_ID -c class_value')
  JOBUSR(TWS_user_name)
```

To define and submit an SAP job dynamically, use the following syntax:

Job definition syntax

-job *job_name* [{ **-i | -id** } *job_ID*] **-flag type=exec** [{ **-host | -ts** } *host_name*] [**-sg** *server_group*] [**-client** *source_client*] [**-exec_client** *target_client*] [**-rfc_client** *rfc_logon_client*] [**-c** *class_value*] [**-bdc_job_status_failed** *bdc_processing*] [{ **-nobdc | -nobdcwait** } [**-bapi_sync_level** { **high | medium | low** }] [**-s** *starting_step_number*]] [**-s** *Step_number attribute_name* [= *attribute_value*]] [**-v** *Step_number variant_name*] [**-vtxt** *Step_number variant_description*] [**-vpar** *Step_number* **name** = *variant_value*] [**-vsel** *Step_number* **name** = { **i | e** } # *operation* # *lowest* [# *highest*]] [**-vtemp** *Step_number*] [**-recipient** *R/3_login_name*] [**-rectype** *recipient_type*] [**-flag** { **reccp | recbl** }] [**-flag recex**] [**-flag recnf**] [**-flag** { **im | immed** }] [**-flag** { **enable_applinfo | disable_applinfo** }] [**-flag** { **enable_appl_rc | disable_appl_rc** }] [**-flag** { **enable_joblog | disable_joblog** }] [**-flag** { **enable_job_interceptable | disable_job_interceptable** }] [**-flag** { **enable_spoollist | disable_spoollist** }] [**-flag pc_launch**] [{ **-debug** }] [**-tracelvl** { **1 | 2 | 3** }] [{ **-rfctrace** }] [**-rfc_client** *rfc_logon_client*]

The following is an example of a definition for the SAPTEST job:

```
-job SAPTEST -C A -s1 program=BTCTEST -s1 type=A -s1 pr_release
-s2 report=BTCTEST -s2 variant=BVT -s2 type=A -flag type=exec
-vpar2 TESTNAME=test -vtxt2 Test
```

Table 19: Task string parameters for SAP jobs (dynamic definition) on page 130 describes the parameters for the task string to define SAP jobs dynamically.

Note: The parameter values are case sensitive.

Table 19. Task string parameters for SAP jobs (dynamic definition)

Section	Parameters	Description
JOB	-job *job_name*	The name of the job to be run. This parameter is mandatory.
	-i *job_ID* -id *job_ID*	The unique SAP job ID. Specify this parameter if you are submitting a job that refers to a predefined job template stored in the SAP database for which you want to change a parameter.
	-host *host_name* -ts *host_name*	The name of the SAP workstation where the job is to be run. *host_name* has the format *hostname_SAPsystemname_SAPsystemnumber*. For example, the name of a host might be amss80a0_gs7_90 These parameters are mutually exclusive with -sg.
	-sg *server_group*	The name of the SAP server group where the job is to be run. Use this parameter to run the job on an application server

Table 19. Task string parameters for SAP jobs (dynamic definition)

(continued)

Section	Parameters	Description
		that belongs to the group. The server group must exist on the SAP system, otherwise an error code is returned and the job is not launched. This parameter is case-sensitive and can be up to 20 characters. It is mutually exclusive with -host and -ts.
	-client *source_client*	The number that identifies the SAP client where the job definition is to be found, regardless of the client number defined by the **r3client** key in the options file. This parameter has no effect if a job ID is specified in the job definition.
	-exec_client *target_client*	The number that identifies the SAP client where the job is to be run, regardless of the client number defined by the `r3client` key in the options file. This parameter requires that the client-dependent data (such as the user name and report variants) exists on both the source and target clients. This parameter has no effect if a job ID is specified in the job definition.
	-rfc_client *rfc_logon_client*	The number that identifies the SAP client to be used for RFC logon. This value overwrites the value specified by the r3client keyword in the corresponding r3batch options file.
	-c *class_value*	The priority with which the job runs in the SAP system. Possible values are: **A** High priority **B** Medium priority **C** Low priority. This is the default value.
	-flag type=exec	Specify this parameter to enable the dynamic definition of the SAP job. This parameter is mandatory.

Table 19. Task string parameters for SAP jobs (dynamic definition)

(continued)

Section	Parameters	Description
JOB	-bdc_job_status_failed *bdc_processing*	How IBM Workload Scheduler sets the completion status of a job running BDC sessions, according to a possible BDC processing failure. The allowed values are: ***n*** If at least *n* BDC sessions failed (where *n* is an integer greater than 0), IBM Workload Scheduler sets the job completion status as failed. **all** If all the BDC sessions failed, IBM Workload Scheduler sets the job completion status as failed. **ignore** When all the BDC sessions complete, regardless of their status, IBM Workload Scheduler sets the job completion status as successful. This is the default value. If **-nobdc** or **-nobdcwait** is set, this option is ignored.
	-nobdc -nobdcwait	Disables the BDC Wait option (enabled by default) to have the job considered as completed even if not all its BDC sessions have ended.
	-bapi_sync_level	Specifies the synchronization level between the SAP function modules BAPI_XBP_JOB_COPY and BAPI_XBP_JOB_START_ASAP. Allowed values are: **high** All RFC calls between BAPI_XBP_JOB_START_ASAP and BAPI_XBP_JOB_COPY are synchronized. This is the default. **medium** The RFC calls to BAPI_XBP_JOB_START_ASAP are synchronized.

Table 19. Task string parameters for SAP jobs (dynamic definition)

(continued)

Section	Parameters	Description
		low The RFC calls are not synchronized.
STEP	-s *starting_step_number*	The number of the starting step.

	-s*step_number attribute_name=attribute_value*	The step number and its attributes, where:
		step_number
		The number of the step being defined. Each step is identified by a sequential number (1, 2, 3, ...*n*) using the *step number*.
		attribute_name
		The name of the attribute.
		attribute_value
		The value of the attribute. It is optional for some attributes.
		Attributes can be defined in any order, but cannot be repeated for the same step. Attribute validation is performed before the job is created in the SAP system. If the validation fails, the IBM Workload Scheduler job goes into the ABEND state. For a detailed description of each attribute and its values, see Defining attributes for ABAP steps on page 139 and Defining attributes for external programs and external commands steps on page 142.
		For example, the following step (step 8) is an ABAP module running the report "MYPGM" and has two attributes, only one of which has a value.
		`-s8 type=A -s8 program=MYPGM`
		`-s8 pr_cover="My title" -s8 pr_immed`

VARIANT[1]	-v*step_number* name	The variant name for the specified step number.
	-vtxt*step_number variant_description*	The textual description of the variant, in the IBM Workload Scheduler logon language (customizable with the `TWSXA_LANG` option of `r3batch`). The maximum length is 30 characters. Not valid for temporary variants.

Table 19. Task string parameters for SAP jobs (dynamic definition)

(continued)

Section	Parameters	Description
	-vpar*step_number* name=*value*	For ABAP modules only. The value for a variant parameter for the specified step number. This parameter is mandatory when creating a new variant. For a complete list of the supported attributes for ABAP steps, see Defining attributes for ABAP steps on page 139.
	-vsel*step_number* name=*sign#operation#lowest*[*#highest*]	For ABAP modules only. The value for a variant selection option for the specified step number. **sign** Sign of the operation. Possible values are: **I** Include **E** Exclude **operation** Possible values are: **EQ** Equals **NE** Not equal to **BT** Between **NB** Not between **LT** Less than **LE** Less than or equal to **GT** Greater than

Table 19. Task string parameters for SAP jobs (dynamic definition)

(continued)

Section	Parameters	Description
		GE Greater than or equal to **CP** Contains pattern **NP** Does not contain pattern **lowest** Low value of the selection. You can use up to 45 characters. **highest** High value of the selection. You can use up to 45 characters. This attribute is optional. For a complete list of the supported attributes for ABAP steps, see Defining attributes for ABAP steps on page 139.
	-vtemp*step_number*	For ABAP modules only. Specifies to assign a temporary variant to the specified step number. Temporary variants are created ad-hoc by the SAP system and assigned to the job instance when it is run. The lifecycle of the temporary variant is determined by the SAP system. If the job is deleted by SAP, then the temporary variant is deleted. See Examples: Dynamically defining and updating SAP jobs on page 144 to refer to examples that demonstrate the behavior of temporary variants.
SPOOL	-recipient *name*	The login name of an SAP user.
	-flag {reccp|recbl}	Specifies how the spool list is sent to the recipient. Possible values are: **reccp** The spool list is sent as a copy. **recbl** The spool list is sent as a blind copy.

Table 19. Task string parameters for SAP jobs (dynamic definition)

(continued)

Section	Parameters	Description
	-flag recex	Specifies that the spool list is sent as an express message to the recipient.
	-flag recnf	Specifies that the recipient is not allowed to forward the spool list.
	-rectype *type*	Specifies the recipient type. Possible values are: **blank** SAP user (default value) **B** SAP user **C** Shared distribution list **D** X.500 address **G** Organization object/ID **H** Organization unit **I** SAP object **L** Telex number **O** SAPoffice user **P** Private distribution list **R** SAP user in another SAP system **U** Internet address

Table 19. Task string parameters for SAP jobs (dynamic definition)

(continued)

Section	Parameters	Description
		1 Other recipient type
FLAGS	-flag im -flag immed	Specifies to launch the job immediately, meaning that if there are no spare work processes, the job fails.
	-flag enable_applinfo -flag disable_applinfo	Enables or disables the retrieval and appending of the SAP application log to the stdlist of IBM Workload Scheduler.
	-flag enable_appl_rc -flag disable_appl_rc	Enables or disables the mapping of the SAP application return code to the IBM Workload Scheduler return code. The SAP application return code is mapped only if **-flag enable_applinfo** is set and the application log contains the application return code.
	-flag enable_joblog -flag disable_joblog	Enables or disables retrieval of the joblog.
	-flag enable_joblog -flag disable_joblog	Enables or disables retrieval of the joblog.
	-flag enable_job_interceptable -flag disable_job_interceptable	Enables or disables the job launched by r3batch to be intercepted by SAP. If enabled, when r3batch launches a job and the SAP job interception feature is enabled, the job can be intercepted if it matches previously defined criteria. If disabled, the job launched by r3batch cannot be intercepted by SAP. This setting overwrites the setting in the common options file.
-flag pc_launch	Specifies to launch child jobs that are in scheduled state. **ON** The product launches child jobs that are in scheduled state.	

Table 19. Task string parameters for SAP jobs (dynamic definition)

(continued)

Section	Parameters	Description
	OFF The product does not launch child jobs that are in scheduled state. This is the default value. **Note:** You can use this option only if you activated the parent-child feature on the SAP system. On the XBP 2.0 (or later) SAP system, you activate this feature by using the INITXBP2 ABAP report.	
TRACING	-debug	Enables maximum trace level.
	-tracelvl 1|2|3	Specifies the trace setting for the job. Possible values are: **1** Only error messages are written in the trace file. This is the default. **2** Informational messages and warnings are also written in the trace file. **3** A most verbose debug output is written in the trace file. For more details, refer to Configuring the tracing utility on page 54.
	-rfctrace -trace	Enables RFC trace.

Note: See Examples: Dynamically defining and updating SAP jobs on page 144 to refer to examples that demonstrate the behavior of variants and temporary variants.

1. The following rules apply when you create or update SAP jobs dynamically:
   - To create or reference a variant within an ABAP step, you can use one of the following equivalent syntaxes:
     - -s1 Variant=Var1
     - -s1 Parameter=Var1
     - -v1 Var1
   - If a variant does not exist, it is created with the parameters specified in the job definition statement. In this case, all the required attributes of the variant must be given a value. You cannot create empty variants. For example, if you specify **-vtemp1**, with no value assigned, an empty temporary variant is erroneously created.
   - If a variant is already present in the SAP system, its values are modified according to the command line parameters. If the existing variant is an extended one, a new instance of it is created with resolved placeholders and updated counters. This new variant instance is then updated using the values from the command line. Finally, the job step is run using this variant instance.
   - All changes to the variant values are permanent. That is, IBM Workload Scheduler neither restores the old values of the variants, nor deletes the variants created after the job is run. IBM Workload Scheduler does not change the case of the variant values.

Defining attributes for ABAP steps

To create and submit SAP jobs dynamically, look at the table and define the attributes for ABAP steps.

Table 20: Supported attributes for ABAP step definition on page 139 shows a complete list of the supported attributes for ABAP step module definition:

Table 20. Supported attributes for ABAP step definition

Attribute name	Synonym	Description	Required
type	typ	Specify the step type. Possible values are: • A • ABAP The product performs a check for correct attribute values prior to launching the job.	✓
program		Specify the ABAP program name.	✓
parameter		Specify the ABAP variant name.	✓
user	authcknam	Specify the user of the step.	✓
language	lang	Specify the step language.	✓

Table 20. Supported attributes for ABAP step definition (continued)

Attribute name	Synonym		Description	Required
			This attribute accepts language names in either the ISO format (two characters, for example DE, EN) or the R/3 format (one character, for example D, E). If this attribute is not specified, the login language of the access method is used (customize using the option **twsxa_lang** in the r3batch options files). The product performs a check for a valid language prior to launching the job.	
pr_dest	printer	pdest	Print Parameter: Specify the printer for the output.	
pr_copies	prcop		Print Parameter: Specify the number of copies. The value of this attribute must be numeric. A corresponding check is performed prior to launching the job.	
pr_lines	linct		Print Parameter: Specify the page length. The value of this attribute must be numeric. A corresponding check is performed prior to launching the job.	
pr_columns	linsz		Print Parameter: Specify the page width. The value of this attribute must be numeric. A corresponding check is performed prior to launching the job.	
pr_auth	prber		Print Parameter: Authorization	
pr_arcmode	armod		Print Parameter: Archiving mode	
pr_sapbanner	prsap		Print Parameter: SAP cover page	
pr_exp	pexpi		Print Parameter: Spool retention period The value of this attribute must be a single digit. A corresponding check is performed prior to launching the job.	
pr_recip	prrec		Print Parameter: Recipient	
pr_spoolname	plist		Print Parameter: Name of spool request [1]	
pr_format	paart		Print Parameter: Print formatting [1]	

Table 20. Supported attributes for ABAP step definition (continued)

Attribute name	Synonym	Description	Required
pr_dep	prabt	Print Parameter: Department on cover page [1]	
pr_spoolds	prdsn	Print Parameter: Name of spool data set [1]	
pr_spoolprio	priot	Print Parameter: Spool request priority [1]	
pr_immed	primm	Print Parameter: Print immediately [2]	
pr_release	prrel	Print Parameter: Delete after printing [2]	
pr_banner	prbig	Print Parameter: Selection cover page [2]	
pr_newspool	prnew	Print Parameter: New spool request [1] [2]	
pr_cover	prtxt	Print Parameter: Text for cover page [1]. If the string contains spaces it must be enclosed between single quotes.	
pr_hostcover	prunx	Print Parameter: Host spool cover page [1] . Possible values are: **' '** Blank. Does not use any cover page. **'X'** Prints the host cover page. **'D'** Prints the default host cover page.	
al_sapobject	sap_object	SAP ArchiveLink: Object type of business object	
al_object	object	SAP ArchiveLink: Document type	
al_info	info	SAP ArchiveLink: Info field	
al_id	archiv_id	SAP ArchiveLink: Target storage system [1]	
al_doctype	doc_type	SAP ArchiveLink: Document class [1]	
al_rpchost	rpc_host	SAP ArchiveLink: PRC host [1]	
al_rpcserv	rpc_servic	SAP ArchiveLink: RPC service / RFC destination [1]	
al_iface	interface	SAP ArchiveLink: Name of communication connection component [1]	
al_client	mandant	SAP ArchiveLink: Client [1]	
al_report		SAP ArchiveLink: Report name [1]	
al_text	arctext	SAP ArchiveLink: Text information field [1]	
al_date	datum	SAP ArchiveLink: Archiving date [1]	

Table 20. Supported attributes for ABAP step definition (continued)

Attribute name	Synonym	Description	Required
al_user	arcuser	SAP ArchiveLink: Data element for user [1]	
al_printer		SAP ArchiveLink: Target printer [1]	
al_format	formular	SAP ArchiveLink: Output format [1]	
al_path	archivpath	SAP ArchiveLink: Standard archive path [1]	
al_protocol	protokoll	SAP ArchiveLink: Storage connection protocol [1]	
al_version		SAP ArchiveLink: Version number [1]	

Note:

1. This attribute is available for BC-XBP 2.0 and later.
2. This attribute is a flag, that is, it does not have a value, for example: `-s2 pr_release`.

IBM Workload Scheduler performs the following syntax validation on job attributes:

- Only valid attributes are allowed.
- Checks if a particular attribute requires a value.
- The values of the following attributes are checked:
  - type
  - language
  - pr_copies
  - pr_lines
  - pr_colums

Validation is performed before the job is created in the SAP system. If the validation fails, the IBM Workload Scheduler job goes into the ABEND state.

Defining attributes for external programs and external commands steps

Table 21: Supported attributes for external programs and external commands step definition on page 142 shows a complete list of the supported attributes for external programs and external commands step definition.

Table 21. Supported attributes for external programs and external commands step definition

Attribute name	Synonym	Description	Required
type	typ	The step type can assume one of the following values:	✓

Table 21. Supported attributes for external programs and external commands step definition (continued)

Attribute name	Synonym	Description	Required
		For external programs • X • EXTPRG **For external commands** • C • EXTCMD Before launching the job, the product performs a check for correct attribute values.	
report		ABAP program name or name of the external program or command.	✓
parameter		Parameters for the external program or command.	
user	authcknam	User of the step.	
language	lang	Step language. This attribute accepts language names in either the ISO format (two characters, for example DE, EN) or the R/3 format (one character, for example D, E). If this attribute is not specified, the login language of the access method is used (customize using the **twsxa_lang** option in the `r3batch` option files). The product performs a check for a valid language prior to launching the job.	
targethost	xpgtgtsys	Target host for the external program or command. This name must be exactly the same as the name shown in the External Operating System Commands table in the SAP system (transaction `sm69`).	
os	opsystem	Operating system for the external command. This name must be exactly the same as the name shown in the External Operating System Commands table in the SAP system (transaction `sm69`).	
termcntl	waitforterm	Control flag: if an external command or program is to be run synchronously. [2]	
tracecntl		Control flag: if SAP tracing level 3 is activated for tracing SAPXPG, the program that starts an external command or program. [1] [2]	

Table 21. Supported attributes for external programs and external commands step definition (continued)

Attribute name	Synonym	Description	Required
stdoutcntl		Control flag: indicates if standard output from an external command or program is to be written to the job log. [1] [2]	
stderrcntl		Control flag: indicates if standard error from an external command or program is to be written to the job log. [1] [2]	

Note:

1. This attribute is available for BC-XBP 2.0 and later.
2. This attribute is a flag, that is, it does not have a value, for example: `-s2 pr_release`.

IBM Workload Scheduler performs the following syntax validation on job attributes:

- Only valid attributes are allowed.
- Checks if a particular attribute requires a value.
- The values of the following attributes are checked:
  - type
  - language
  - pr_copies
  - pr_lines
  - pr_colums

Validation is performed before the job is created in the SAP system. If the validation fails, the IBM Workload Scheduler job goes into the ABEND state.

Specifying job parameters using variable substitution

Parameters can be provided at run time using the variable substitution feature. For example, the value appears as:

```
-s1 report=&VARNAME
```

The variable substitution process occurs while IBM Workload Scheduler is creating the symphony file.

Examples: Dynamically defining and updating SAP jobs

This section describes some usage examples of this feature:

Job definition and run scenario using the -flag type=exec parameter

The following example creates and runs a 3-step job. The first step runs the ABAP MYPROG1 using variant VAR01 and associated variant parameter. Step 2 has a step user defined. Step 3 uses the same ABAP as step 1 with no associated variant.

The only requirement is that the elements referred to are known in the SAP system (user, program). If the variant does not exist, there should be a set of values to define the content of the variant for its creation (pairs of `-vparN -vselN` parameters for the parameters and selections of the ABAP program).

```
-job TESTJOB01 -c A
-s1 type=A -s1 program=MYPROG1
-v1 VAR01 -vpar1 TESTNAME=TST
-s2 report=SPOOLX1 -s2 user=PRTUSER
-s3 type=A -s3 program=MYPROG1 -flag type=exec
```

The job returns job ID 12345678

Job copy and overwrite the job created in the previous step

The following job statement references the job created in the previous example. A new copy of the job is made and the parameters specified in the invocation are used to update the definition. In this case the variant for step 1 is modified and a new external program step (Step 4) is added.

```
-job TESTJOB01 -i 12345678
-s1 variant=VAR01A
-vpar1 TESTNAME=TST2
-s4 type=X -s4 report=niping -s4 parameter=-t
-flag type=exec
```

Copy and overwrite a job referencing an existing job template

The following example shows a job creation referencing a job template (previously created without using this feature). A template job called TEMPLAJOB already exists on the SAP system with an ID of 56780123. It is a single ABAP step job to which we now add some print parameters.

```
-job TEMPLAJOB
-I 56780123 -s1 pr_immed
-flag type=exec
```

A temporary variant is created using the information indicated in the expression

The following is the syntax to be used:

```
-vpar1 <parameter_name>=<parameter_value> ...
-vsel1 <selection_option_name>
... -vtemp1
```

The following example shows how you can submit a job that creates a temporary variant that is assigned to step number 1, and assigns a value to a variant parameter for step number 1:

```
-job TESTJOB01 -C A -flag type=exec -user R3USER
-s1 type=A -s1 program=MYPROG1
-vtemp1 -vpar1 TESTNAME=TST
```

The following example shows how you can submit a job that creates a temporary variant that is assigned to step number 1, assigns a value to a variant parameter for step number 1, and assigns a value to a variant selection option (date) for step number 1:

```
-job TESTJOB01 -C A -flag type=exec -user R3USER
-s1 type=A -s1 program=MYPROG1
-vtemp1 -vpar1 FILENAME=FLN
-vsel1 date=E#BT#20110101#20110412
```

Assign a temporary variant to the specified step number

The following is the syntax to be used:

```
-v1 <temporary_variant_name> -vtemp1
```

The following is an example of how you can submit a job to assign a temporary variant, which has already been created (as in the previous example), and assign a value to step number 1:

```
-job TESTJOB01 -C A -flag type=exec -user R3USER
-s1 type=A -s1 program=MYPROG1
-vtemp1 -v1 &000000000001
```

The value for a temporary variant that already exists is substituted with the value indicated in the expression

The following is the syntax to be used:

```
-v1 <temporary_variant_name> -vpar1 <parameter_name>=<parameter_value> ...
-vsel1 <selection_option_name> ... -vtemp1
```

The following is an example of how you can submit a job that substitutes the value of a temporary variant, which must already exist, with a new value. The temporary variant must exist, otherwise, the expression returns an error.

```
-job TESTJOB01 -C A -flag type=exec -user R3USER
-s1 type=A -s1 program=MYPROG1
-vtemp1 -v1 &000000000001 -vpar1 TESTNAME=TST2
```

Defining conditions and criteria

IBM Workload Scheduler accesses the Computer Center Management System (CCMS) Background Processing components of SAP systems through the BC-XBP interface to provide additional capabilities from the Dynamic Workload Console, one of those being the Criteria Manager.

IBM Workload Scheduler supports the BC-XBP 3.0 interface which provides functions to control R/3 batch jobs.

The Criteria Manager is a tool that enables you to define conditions and criteria that, when combined, form complex dependencies that you can use in the following contexts:

- Managing raised events in the SAP event history.
- Managing reorganization tasks against the SAP event history.
- Intercepting jobs.

If you have other types of criteria defined on your SAP system, then you can perform other actions in addition to those listed in this section.

The criteria profile

The Criteria Manager enables you to define a criteria profile which is a container for a combination of criteria. The criteria profile can be of various types and each criteria type has a standard set of selection criterion. For each criteria, you can specify a single value, a range of values by indicating a lower and upper limit, and multiple values. The following is the standard set of selection criterion for each criteria profile type. In addition to these, you can also see any other types of criteria profiles you have defined on your SAP system:

Event History

EVENTID

The identifier of the event defined in the SAP system.

EVENTPARM

The parameter of the event defined in the SAP system.

PARMID

The identifier of the parameter of the event defined in the SAP system.

Event History Reorg

Event State

The state of the event.

Event Timestamp

The timestamp for the event.

Interception

Job Name

A name identifying the job.

Job Class

The class assigned to the job that represents the priority with which the job runs in the SAP system.

The criteria hierarchy

You create and combine criteria in a criteria hierarchy. The criteria hierarchy is a set of all the criteria that must be fulfilled for a specific action to take place in the specific context. For example, you can define a criteria hierarchy to log all raised events in the SAP event history with an event name that begins with "CRITICAL_EVENT" and with an event argument equal to 150.

The criteria in the hierarchy is grouped in nodes and relationships between the nodes are determined by the logical operators AND or OR. You can nest nodes in other nodes.

To have the criteria profile begin processing, the criteria profile must be activated. Only one criteria profile of the same type can be active at one time.

Example

An example

See Example: Defining which raised events to log on page 148 for an example that demonstrates how to build a criteria hierarchy to manage the logging of raised events in the SAP event history.

Example: Defining which raised events to log

The event history stores all events that are raised by the system. You can define specific criteria so that only raised events that match certain criteria are logged.

The event history enables IBM Workload Scheduler to consume events that are raised by the SAP system.

Checking the log of raised events gives you access to the following information:

- Verify that an event was raised in the system.
- Verify if the event was processed.

In the example that follows, an event history criteria profile is created that contains the definition of the criteria, the criteria hierarchy, that events must fulfill to be logged in the event history. The criteria profile must then be activated so that it can begin processing events according to the criteria.

The criteria profile, `Event profile 1`, contains a criteria hierarchy that logs only those events in the event history with event name that begins with `CRITICAL_EVENT` and event argument equal to "789".

Create the criteria profile

A criteria profile contains the definition of the criteria you want to set for logging raised events.

About this task

Create a criteria profile, `Event profile 1`, of type, Event History, to contain the criteria hierarchy.

1. In the Navigation bar at the top, click **Design > SAP > Manage SAP Criteria Profiles**.
2. In **Engine name**, select the name of the IBM Workload Scheduler engine connection from which you want to work with SAP jobs.
3. In **Workstation name**, type the name of the workstation where the SAP job runs. This is the workstation with the r3batch access method that communicates with the remote SAP system. If you do not know the name of the workstation, click the **Lookup Workstations** icon to enter your filter criteria and click **Search**. If you enter a string representing part of the workstation name, it must be followed by the asterisk (*) wildcard character. Both the question mark (?) and asterisk (*) are supported as wildcards. You can also simply use the asterisk wildcard character (*) to display all workstations. Optionally, specify any of the other search criteria available and click **Search**.
   Result
   From the results displayed, select the workstation and click **OK**.
4. In **Options file**, specify an options file that resides on the specified workstation. Each workstation can have one or more options files that can be used to customize the behavior of the r3batch access method, except for extended agent workstations, where only one options file can exist and therefore does not need to be specified. For the workstation specified, enter the file name of the options file or click pick icon to search for options files that reside on the specified workstation and select one.
5. Click **Go**.
6. From the Criteria Manager main view, click **New** to create a criteria profile.
7. Select **Event History** as the type of criteria profile you want to create.

8. Enter descriptive text that enables you to easily identify the criteria profile in the table of criteria profiles. Type `Event profile 1`. Avoid using special characters such as, < (less than), > (greater than), or the ' (apostrophe) in this field.
9. Click **Save**.

Results

The criteria profile is displayed in the list of criteria profiles and it is not yet active.

What to do next

Next, begin building the criteria hierarchy. The criteria profile is the container for the criteria hierarchy.

Build the criteria hierarchy

The criteria hierarchy is stored in the criteria profile and is made up of criteria. A group of criteria is contained in a node.

Before you begin

The criteria hierarchy is made up of a combination of nodes and criteria. A node contains a group of criteria where the relationship between the criteria is determined by an AND or an OR relation. You can nest nodes in other nodes. By default, a top level AND node is created in the criteria hierarchy. You can create other AND or OR nodes nested in this parent node. You can also add one or more criterion to the nodes. Add an AND node when all of the criteria defined in the node must be fulfilled. Add an OR node when at least one of the criteria defined in the node must be fulfilled.

About this task

In this example, define a criterion that logs all events whose name begins with `CRITICAL_EVENT` and with event argument equal to `789`.

1. Click  to create a new criterion in the default AND node.
2. In **Description**, type `Criterion 1`.
3. In **EVENTID**, click  to specify the value for the EVENTID field.
4. Leave the default value **Select** to indicate to use the selection criterion specified when processing events.
5. In **Options**, select **Pattern** and in **Single Value or Lower Limit**, type `CRITICAL_EVENT*`.
   Result
   This sets the condition for the event name.
6. In **EVENTPARM**, click  to specify the value for the EVENTPARM field.
7. Leave the default value **Select** to indicate to use the selection criterion specified when processing events.
8. In **Options**, select **Equal to** and in **Single Value or Lower Limit**, type `789`.
   Result
   This sets the condition for the event argument.
9. Click **Save** to save the criterion definition.

Results

The criteria profile now contains a criterion that specifies which raised events must be logged. You can continue to create another criteria in the same parent node or you can nest either an AND or an OR node in the parent node to determine the

logical relation between the criteria that the nested node will contain. Add an AND node within which you can create one or more criteria where all the criteria specified in the node must be fulfilled, or add an OR node within which you can create one or more criteria where at least one of the criteria specified must be fulfilled.

What to do next

To apply this criteria profile so that it begins processing events according to the criteria defined, you must activate the criteria profile.

Activate the criteria profile

To apply the `Event profile 1` criteria profile so that it begins processing raised events according to the criteria specified in the criteria hierarchy, you must activate the criteria profile.

About this task

A criteria profile can either be active or not active. For a criteria profile to take effect, the profile must be activated. Only one criteria profile of the same type can be active at one time. Criteria profiles cannot be edited if they are in the active state. Follow the procedure to activate the `Event profile 1` criteria profile.

1. Select the `Event profile 1` criteria profile from the table of criteria profiles.
2. Select **Activate** from the toolbar. 

Results

The status of the criteria profile is updated to show that it is now active. The criteria profile can now begin to process raised events according to the specifications of the criteria hierarchy and log them to the event history. If another criteria profile of the same criteria type was active, its status changes to inactive.

Using the BDC Wait option

By using the Batch Data Collector (BDC) Wait option, you can specify that a SAP job launched by IBM Workload Scheduler is not to be considered complete until all of its BDC sessions have completed.

About this task

The Batch Data Collector (BDC) Wait option prevents other IBM Workload Scheduler jobs that are dependent on the SAP job from being launched until all of the related BDC sessions for the SAP job have ended.

To use the option, a SAP job must write informational messages in its job log. This can be done by modifying the SAP function module BDC_OPEN_GROUP as follows:

```
FUNCTION BDC_OPEN_GROUP.
...
CALL 'BDC_OPEN_GROUP' ID 'CLIENT'      FIELD CLIENT
                      ID 'GROUP'       FIELD GROUP
                      ID 'USER'        FIELD USER
                      ID 'KEEP'        FIELD KEEP
                      ID 'HOLDDATE'    FIELD HOLDDATE
                 ID 'DESTINATION' FIELD DEST
                 ID 'QID'         FIELD QID
                 ID 'RECORD'      FIELD RECORD
```

```
                ID 'PROG'        FIELD PROG.
*
IF SY-SUBRC EQ 0.
  BQID   = QID.
  BUSER  = SY-MSGV1.
  BGROUP = GROUP.
* CALL FUNCTION 'DB_COMMIT'.
  CALL FUNCTION 'ENQUEUE_BDC_QID'
  EXPORTING   DATATYP   = 'BDC '
  GROUPID    = BGROUP
  QID              = BQID
  EXCEPTIONS  FOREIGN_LOCK  = 98
  SYSTEM_FAILURE   = 99.

  IF SY-SUBRC EQ 0.
     message i368(00) with 'BDCWAIT: ' qid.
  ENDIF.

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

Note: The actual parameters of the call of the C function `(CALL 'BDC_OPEN_GROUP' ID ...)` might vary depending on the SAP release. With this approach, you obtain a global change in your SAP system.

The completion status of a SAP job launched by IBM Workload Scheduler is based on the value you set for the **bdc_job_status_failed** option. By default, this option is set to `ignore`, meaning that the job is considered successfully completed when the BDC sessions are finished, regardless of their success or failure. For details about the **bdc_job_status_failed** option, refer to Table 17: Task string parameters for SAP jobs on page 111.

Job interception and parent-child features

This section describes how the job interception and parent-child features of BC-XBP 2.0 and 3.0 are supported by IBM Workload Scheduler.

Distributed **Note:** The process of defining relaunch criteria and collecting and relaunching intercepted jobs is supported only in distributed environments and not in z/OS environments.

Implementing job interception

The high-level steps required to implement job interception.

About this task

Job interception is a feature of both the BC-XBP 2.0 and BC-XBP 3.0 interfaces. It enables IBM Workload Scheduler to have a very sophisticated control over the jobs launched by SAP users from the SAP graphical interface.

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

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

Job interception with the BC-XBP 2.0 interface is based on the single extended agent workstation, whereas with the BC-XBP 3.0 interface, job interception is based on the currently active job interception criteria profile.

Note:

- Jobs launched by IBM Workload Scheduler, or by any other external scheduler using the BC-XBP interface, can be intercepted provided the **job_interceptable** option in the common options file is set to `ON`, and the **-flag enable_job_interceptable** keyword is included in the job definition.
- Ensure that the job interception and job throttling features are not running at the same time. The interception collector jobs fail if a job throttler instance is running. To stop the job throttler, refer to Step 5. Starting and stopping the job throttling feature on page 181.

The following are the high-level steps required to implement job interception for both the BC-XBP 2.0 and 3.0 interfaces.

Job interception and the BC-XBP 2.0 interface

About this task

To set up IBM Workload Scheduler to handle job interception in an SAP environment with the BC-XBP 2.0 interface, implement the following steps:

1. Install the BC-XBP 2.0 interface. Refer to SAP Note 604496 to know if your SAP system already has the BC-XBP 2.0 interface, or which SAP support package you need to install to enable it.
2. Define an IBM Workload Scheduler job to periodically collect the intercepted SAP jobs.
3. Specify interception criteria in the SAP system.
4. Specify interception criteria in IBM Workload Scheduler from the Monitor Workstations portlet on the Dynamic Workload Console. The criteria is set at workstation level.
5. Activate the job interception feature of the BC-XBP 2.0 interface.

Job interception and the BC-XBP 3.0 interface

About this task

To set up IBM Workload Scheduler to handle job interception in an SAP environment with the BC-XBP 3.0 interface, implement the following steps:

1. Verify if the BC-XBP 3.0 interface is already installed on the SAP system.
2. Define an IBM Workload Scheduler job to periodically collect the intercepted SAP jobs.
3. Specify interception criteria in the SAP system.
4. Specify interception criteria in IBM Workload Scheduler from the Manage SAP Criteria Profiles portlet on the Dynamic Workload Console.
5. Activate the job interception feature of the BC-XBP 3.0 interface.

Collecting intercepted jobs periodically for BC-XBP 2.0

With the BC-XBP 2.0 interface, you can configure the job interception collector using an IBM Workload Scheduler job that periodically retrieves intercepted jobs and relaunches them.

About this task

Define an IBM Workload Scheduler job that uses the SAP interception collector task to collect intercepted jobs and restart them.

To define an IBM Workload Scheduler job that collects intercepted job and relaunches them, use the following syntax:

```
ENGINE_NAME_HOSTING_XA#[folder/]JOBNAME
  SCRIPTNAME "TWA_home/methods/r3batch -t HIJ -c UNIQUE_ID"
  DESCRIPTION "Collects intercepted jobs on SAP XA XA_Unique_ID"
  STREAMLOGON TWSuser
  RECOVERY STOP
```

Where:

ENGINE_NAME_HOSTING_XA

The name of the engine workstation hosting the XA workstation with the r3batch access method that communicates with the SAP system.

JOBNAME

Name of the IBM Workload Scheduler job and the folder in which it is defined, if any.

TWA_home

Fully qualified path to your IBM Workload Scheduler installation.

XA_Unique_ID

The unique identifier for the extended agent workstation. See UNIQUE_ID on page 21 for more details about retrieving the unique identifier.

–t HIJ

This is the SAP task type to run the job interception collector. HIJ stands for `Handle Intercepted Jobs`.

TWSuser

Name of the IBM Workload Scheduler user that launches the access method.

The interception collector job runs at periodical intervals; for example, every 10 minutes. It retrieves all the jobs that have been intercepted since the last run of the interception collector, and launches them again according to a template.

Collecting intercepted jobs periodically for BC-XBP 3.0

With the BC-XBP 3.0 interface, you can configure the job interception collector using an IBM Workload Scheduler job that periodically retrieves intercepted jobs and relaunches them.

About this task

Because intercepted jobs remain in the `Released` and then `Intercepted` status until they are relaunched, you need to use the SAP interception collector task to collect and relaunch them.

To use the job interception collector BC-XBP 3.0 to collect and restart jobs, create the **r3batch_icp** folder in the *TWS_home*`/methods/` path with the correct access rights.

To define an IBM Workload Scheduler job that collects and relaunches jobs use the following syntax:

```
ENGINE_NAME_HOSTING_XA#[folder/]JOBNAME
DOCOMMAND "TWA_home/methods/r3batch -t HIJ -c UNIQUE_ID -- \
   "-profile_id profile_ID_number\""
STREAMLOGON TWSuser
DESCRIPTION "Collects intercepted jobs on SAP XA XA_Unique_ID"
TASKTYPE UNIX
RECOVERY STOP
```

where,

ENGINE_NAME_HOSTING_XA

The name of the engine workstation hosting the XA workstation with the r3batch access method that communicates with the SAP system.

JOBNAME

Name of the IBM Workload Scheduler job and the folder within which it is defined, if any.

TWA_home

Fully qualified path to your IBM Workload Scheduler installation.

XA_Unique_ID

The unique identifier for the extended agent workstation. See UNIQUE_ID on page 21 for more details about retrieving the unique identifier.

–t HIJ

This is the SAP task type to run the job interception collector. HIJ stands for `Handle Intercepted Jobs`.

- profile_id *profile_ID_number*

Specifies the identification number of the interception criteria profile on the SAP system for XBP 3.0.

TWSuser

Name of the IBM Workload Scheduler user that launches the access method.

The interception collector job runs at periodical intervals; for example, every 10 minutes. It retrieves all the jobs that have been intercepted since the last run of the interception collector, and launches them again according to a template.

Note: If the interception collector is configured for XBP 3.0 job interception, but the XBP 2.0 interface is configured on the SAP system, the collector fails. Ensure the XBP interface versions are synchronized.

Setting interception criteria on the SAP system

About this task

In SAP, the interception criteria are held in table `TBCICPT1`. Only jobs that match the criteria of this table are intercepted, when their `start` conditions are fulfilled. All the other jobs are run normally.

You can maintain the entries in this table by using transaction `se16` and setting the following:

- Client number
- Job mask
- User mask

Setting interception criteria on IBM Workload Scheduler

About this task

In IBM Workload Scheduler, interception criteria are defined and used by setting:

Table criteria

For BC-XBP 2.0

You use the **Monitor Workload** of Dynamic Workload Console to set table criteria.

For details about how you set table criteria, see Setting SAP table criteria on the extended agent workstation on page 155.

For BC-XBP 3.0

You set table criteria from the **Administration > Workload Design > Manage SAP Criteria Profiles** panel from the Dynamic Workload Console.

For details about how you set table criteria, see Setting SAP criteria in the job interception criteria profile on page 157.

Template files (optional)

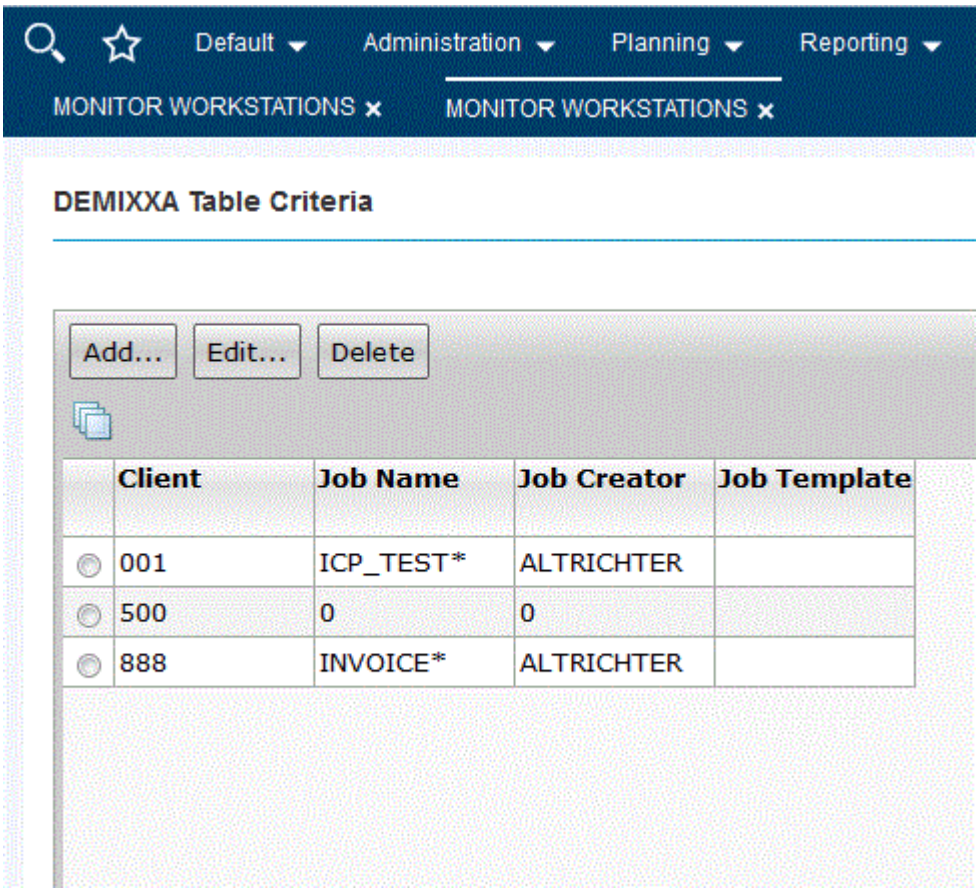
For details about how you create template files, see Using template files on page 158.

Setting SAP table criteria on the extended agent workstation

About this task

To set table criteria with the BC-XBP 2.0 interface on an SAP job using the **Monitor Workload** of the Dynamic Workload Console, follow these steps:

1. Log in to the Dynamic Workload Console.
2. From the **Monitoring and Reporting** menu, click **Orchestration Monitor**.
3. In the **Monitor Workload** window select the engine, enter **Workstation** in the **Object Type** field, and select the plan to display the list of workstations you want to monitor. Click **Run**.
4. Select an extended agent workstation in the table of displayed workstations, and click **More Actions > Table Criteria...** from the toolbar.
5. The **Table Criteria** panel displays. From this panel you can add, delete, edit, or refresh criteria.
   Figure 9. The Table Criteria panel



6. Specify the criteria:
   a. In **Client**, specify the client workstation of the SAP job.
   b. In **Job Name**, specify a filter to match a set of SAP jobs. Use the asterisk (*) wildcard character to match a set of jobs.
   c. In **Job Creator**, specify a filter to match a set of SAP job creator. Use the asterisk (*) wildcard character to match a set of jobs.

d. Optionally, in **Job Template**, specify the template file that contains instructions for the interception collector about how to run the intercepted SAP job under control of IBM Workload Scheduler. For more information about template files, see Using template files on page 158.
e. In **Job Class**, specify the class assigned to the job that represents the priority with which the job runs on the SAP system.

7. Click **OK**.

Setting SAP criteria in the job interception criteria profile

Setting criteria to intercept jobs and relaunch them.

About this task

To set the criteria that defines which SAP jobs to intercept and relaunch with the BC-XBP 3.0 interface using the Dynamic Workload Console, perform the following steps:

1. In the Navigation bar at the top, click **Design > SAP > Manage SAP Criteria Profiles**.
2. In **Workstation name**, type the name of the workstation where the SAP job runs. This is the workstation with the r3batch access method that communicates with the remote SAP system. If you do not know the name of the workstation, click the **Lookup Workstations** icon to enter your filter criteria and click **Search**. If you enter a string representing part of the workstation name, it must be followed by the asterisk (*) wildcard character. Both the question mark (?) and asterisk (*) are supported as wildcards. You can also simply use the asterisk wildcard character (*) to display all workstations. Optionally, specify any of the other search criteria available and click **Search**.
   Result
   From the results displayed, select the workstation and click **OK**.
3. From the Criteria Manager main view, click **New** to create a criteria profile.
4. Select **Interception** as the type of criteria profile you want to create.
5. Enter descriptive text that enables you to easily identify the criteria profile in the table of criteria profiles. Avoid using special characters such as, <(less than), > (greater than), or the ' (apostrophe) in this field.
6. Click **Save**.
   Result
   The criteria profile is displayed in the list of criteria profiles and it is not yet active.
7. On the **Details** tab in the upper-right pane, define the criteria that intercepted jobs must match. For example, to intercept jobs with a job name beginning with "ICP", specify the following criteria:

   a. Click  to define a new criterion.

   b. In **Description**, type `Criterion 1`.

   c. In **JOB NAME**, click  to specify the value for the JOB NAME field.

   d. Leave the default value **Select** to indicate to use the selection criterion specified when intercepting jobs.

   e. In **Options**, select **Pattern** and in **Single Value or Lower Limit**, type `ICP*`.

Result
This sets the condition for the job name.

f. Click **Save** to save the criterion definition.

8. Define the criteria that must be matched to relaunch intercepted jobs. Click the **Job Relaunch Criteria** tab.
   a. Click  to define a new criteria that determines which jobs are relaunched.
   b. In **Client**, specify the client workstation of the SAP job.
   c. In **Job Name**, specify a filter to match a set of SAP jobs. Use the asterisk (*) wildcard character to match a set of jobs.
   d. In **Job Creator**, specify a filter to match a set of SAP job creator. Use the asterisk (*) wildcard character to match a set of jobs.
   e. Optionally, in **Job Template**, specify the template file that contains instructions for the interception collector about how to run the intercepted SAP job under control of IBM Workload Scheduler. For more information about template files, see Using template files on page 158.
   f. In **Job Class**, specify the class assigned to the job that represents the priority with which the job runs on the SAP system.
9. Click **OK**.
10. You can continue to define more criteria and then save the criteria profile.
11. When you are done defining the criteria, save the criteria profile.
12. Select the criteria profile and then click **Activate** from the toolbar. 

Results
The status of the criteria profile is updated to show that it is now active. The criteria profile can now begin to intercept jobs according to the specifications of the criteria hierarchy and relaunch them as defined in the IBM Workload Scheduler job. If another criteria profile of the same criteria type was active, its status changes to inactive.

Using template files

About this task

A template is a file with extension `.jdf` located in the same directory as the interception criteria file (*`TWA_DATA_DIR`*`/methods/r3batch_icp`). The template file contains instructions for the interception collector about how to run the intercepted SAP job under control of IBM Workload Scheduler. Its syntax corresponds to the syntax of `docommand` in conman. You can use any text editor to maintain this file. Ensure that the user, LJUser, is able to read and write to this file.

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

```
alias=SAP_$RUN_$JOBNAME_$JOBCOUNT
```

This means that the intercepted SAP jobs are to be restarted immediately, because of the absence of the `at=` job option. Their IBM Workload Scheduler names are composed of the string `SAP_`, the current run number of the interception collector, and the name and ID of the SAP job.

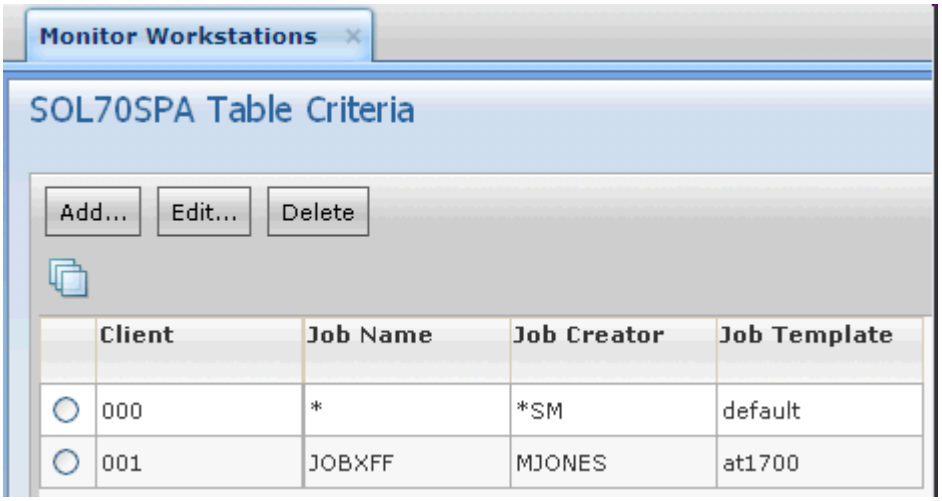
The instruction set for restarting an intercepted SAP job is retrieved in the following order:

1. From the template file, if an existing template is specified in the interception criteria file.
2. From the default template file, if the template is specified in the interception criteria file but does not exist, or if the template is not specified in the interception criteria file.
3. From the default instruction set, if the default template file does not exist.

Job interception example

The following example demonstrates how different template files can be used to determine when an intercepted SAP job is restarted. The interception criteria table contains the following entries:

Figure 10. The Table Criteria panel



The table criteria specified, implies the following:

Client 000

All jobs started in client `000` by SAP users whose user name begins with `sm`, will be intercepted. The interception collector restarts the jobs using the instructions from the default template file `default.jdf`. If the default template file does not exist, then the SAP jobs are restarted immediately as specified in the default instruction set:

```
alias=SAP_$RUN_$JOBNAME_$JOBCOUNT
```

Client 001

The job named, JOBXFF, started in client `001` by SAP user named, MJONES, will be intercepted. The interception collector restarts the jobs using the instructions from the template file `at1700.jdf`. The SAP jobs are restarted at 17:00 with a random name, because of the `alias` command. The template file `at1700.jdf` contains the following entry:

```
alias;at=1700
```

Using placeholders

In the template files you can use a number of placeholders that are replaced by the interception collector at run time. They are listed in Table 22: Placeholders for job interception template files on page 160.

Table 22. Placeholders for job interception template files

Placeholder	Description
$CPU	Name of the extended agent workstation where the interception collector runs.
$CLIENT	Client number of the intercepted SAP job.
$JOBNAME	Name of the intercepted SAP job.
$JOBCOUNT	Job ID of the intercepted SAP job.
$USER	Name of the user who launched the SAP job.
$JOBNUM	Job number of the interception collector.
$RUN	Current run number of the interception collector.
$SCHED	Schedule name of the interception collector.
$RAND	Random number.

The template:

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

instructs the interception collector to restart the SAP job named `DEMO_JOB` with job ID `12345678` on client `100` at `10:00` as IBM Workload Scheduler job `ICP_1432_DEMO_JOB_12345678_100.`

Activating the job interception feature

Activate the job interception feature for the appropriate BC-XBP interface.

About this task

To enable the job interception feature: .

1. Run ABAP report `INITXBP2`.
   Result
   This report shows you the current status of the job interception and parent-child features, and allows you to toggle the status of both features.
2. Select the BC-XBP interface version as appropriate:
   - **Activate 3.0**
   - **Activate 2.0**
3. Save the changes.

The parent-child feature

In some situations, an SAP job dynamically spawns a number of other jobs; for example, to distribute the workload to the free application servers. Prominent examples are the mass activity jobs of the SAP `FI-CA` component. Before BC-XBP 2.0, it was difficult for external schedulers to handle this situation, because the business process does not usually end with the end of the initial job (parent job), but with the end of all subjobs (child jobs).

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

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

When the parent-child feature is active, IBM Workload Scheduler considers an SAP job as finished only after all its child jobs have ended. The status of the IBM Workload Scheduler job remains as `EXEC` while the parent job or any of its child jobs are running.

The status of the IBM Workload Scheduler job becomes `SUCC` if the parent job and all child jobs end successfully. If any of the jobs ended with an error, the status of the IBM Workload Scheduler job becomes `ABEND`.

Note: The parent-child feature can interfere with job interception because, although the parent job cannot be intercepted, any of its child jobs can be intercepted if they match the interception criteria. In this case, the IBM Workload Scheduler job remains in the `EXEC` status until the intercepted child job has been relaunched and has ended.

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

Using Business Information Warehouse

Business Information Warehouse (BIW) is a data warehouse solution tailored to SAP.

Business Information Warehouse (BIW) allows business reporting and decision support.

To use the InfoPackages component, you must have the SAP Business Warehouse Systems, version 2.0B or later installed.

To use the Process Chains component, you must have the SAP Business Warehouse Systems, version 3.0B or later installed.

The Support Package 9 (SAPKW31009) for SAP Business Warehouse version 3.1 is required so that `SAP` can launch process chains.

Business Warehouse components

SAP supports two main Business Warehouse components, InfoPackages and Process Chains.

An InfoPackage is the entry point for the loading process from a specific **InfoSource** (a logical container of data source, generically named **InfoObject**). Technically, an InfoPackage is an SAP job whose aim is to load data. Like any other SAP job, it contains job-pecific parameters such as start time, and dependencies.

A Process Chain is a complex chain of different processes and their relationships. The processes within a process chain are not limited to data load processes, or InfoPackages, but also include:

- Attribute/Hierarchy Change run
- Aggregate rollup
- ABAP program
- Another process chain
- Customer build process

Defining user authorizations to manage SAP Business Warehouse InfoPackages and process chains

What you need to use SAP R/3 Business Warehouse InfoPackages and process chains.

Access method for SAP can manage SAP Business Warehouse InfoPackages and process chains. To use the SAP Business Warehouse functions, you must define an IBM Workload Scheduler user within SAP with full authorization for the ABAP Workbench object S_DEVELOP.

The user must also belong to the following profiles:

- S_BI-WHM_RFC (for Business Information Warehouse version 7.0, or later)
- S_RS_ALL
- Z_MAESTRO

Managing SAP Business Warehouse InfoPackages and process chains

You can manage existing InfoPackages and process chains on SAP systems from SAP.

Business Warehouse InfoPackages and process chains can only be created from the SAP environment. However, the Dynamic Workload Console supports pick lists of InfoPackages and process chains, so that you can also define IBM Workload Scheduler jobs for these existing objects.

You can create IBM Workload Scheduler job definitions that map to SAP jobs that already exist on SAP systems in the following environments:

- Distributed Distributed
- z/OS z/OS

The SAP jobs can run on extended agent workstations, dynamic agent workstations, dynamic pools, and z-centric workstations depending on the type of job definition you choose to create.

This section describes how to perform tasks such as creating the IBM Workload Scheduler job definitions that map to SAP jobs, how to display the details of these jobs, and how to rerun a process chain job.

Creating an IBM Workload Scheduler job that contains InfoPackages or process chains

Creating a job with InfoPackages or process chains.

About this task

This section describes how to create an IBM Workload Scheduler SAP job definition that references a Business Warehouse InfoPackage or Process Chain SAP job.

SAP job definitions can be created using both a distributed or z/OS engine and they can be scheduled to run on the following workstations with the r3batch access method:

- An IBM Workload Scheduler extended agent workstation. A workstation that is hosted by a fault-tolerant agent or master workstation.
- A dynamic agent workstation.
- A dynamic pool.
- A z-centric workstation.

Refer to the Dynamic Workload Console online help for a complete description of all UI elements for both engine types and all supported workstation types.

Take into consideration that:

- To be able to schedule InfoPackages using IBM Workload Scheduler, the scheduling options of the InfoPackage must have:
  - Start type set to **Start later in background process**.
  - Start time set to **Immediate**.
- To be able to control process chains using IBM Workload Scheduler, the scheduling options of the process chain must be **Start Using Meta Chain or API**. If the process chain is set to **Direct Scheduling**, it starts immediately when activated in the SAP system or transported to another SAP system.
- If you are using an operating system that does **not** support Unicode, set the **TWSXA_LANG** option. For details about the operating systems that support Unicode, see Unicode support on SAP on page 76. For details about the **TWSXA_LANG** option, see Setting National Language support options on page 219.

You can create a SAP job definition to reference an InfoPackage or process chain using the Dynamic Workload Console.

The following procedure creates an IBM Workload Scheduler SAP job definition and references an InfoPackage or process chain in the IBM Workload Scheduler database:

1. Click **IBM Workload Scheduler**>**Design**>**Workload Designer**.
2. Select a an engine. The **Workload Designer** is displayed.
3. From the Working List pane, click:
   - z/OS engine: **New > ERP**
   - Distributed engine: **New > Job Definition > ERP**
4. Select the SAP job definition in accordance with the engine and type of agent on which the job runs.

z/OS engine

SAP

This job definition references an existing job on the SAP system and can run on dynamic agent workstations, dynamic pools, and z-centric workstations.

Distributed engine

SAP Job on Dynamic Workstations

This job definition can run on dynamic agent workstations, dynamic pools, and z-centric workstations.

SAP Job on XA Workstations

This job definition can run on extended agent workstations. A workstation that is hosted by a fault-tolerant agent or master workstation.

5. In the Workspace pane, specify the properties for the job definition you are creating using the tabs available. The tabs for each type of SAP job definition are similar, but there are some differences depending on the type of engine you selected and the type of workstation on which the job runs. For more detailed information about the UI elements on each tab, see the Dynamic Workload Console online help.

   The **General** page requires information regarding the workstation that connects to the remote SAP system. If a default SAP connection is already configured, then these fields are alreaday prefilled, otherwise, you can specify the required information on the General page or you can configure a default connection to be used each time it is required in a definition, see Setting the SAP data connection on page 105 for more information.

   On the **Task** page, in **Subtype**, specify either **BW Process Chain** or **BW InfoPackage**.

6. Click **Save** to add the SAP job definition to the IBM Workload Scheduler database.

Task string to define Business Warehouse InfoPackages and process chain jobs

This section describes the task string parameters that control the running of the Business Warehouse InfoPackages and process chain jobs. You must specify them in the following places when you define their associated IBM Workload Scheduler jobs:

- If you use the Dynamic Workload Console, in the **SAP command line** field of the Task page of the SAP job definition panel.
- As arguments of the `scriptname` keyword in the job definition statement, if you use the IBM Workload Scheduler command line.
- As arguments of the `JOBCMD` keyword in the `JOBREC` statement in the `SCRIPTLIB` of IBM Z Workload Scheduler, if you are scheduling in an end-to-end environment.

The string syntax is the following:

Job definition syntax

```
-jobjob_name -i { ipak_ | pchain_ } [ -debug ] [ -trace ] [ -flag { imm | immed } ] [ -flag { enable_pchainlog | disable_pchainlog } ] [ -flag { enable_ipaklog | disable_ipaklog } ] [ -flag { level_all_pchainlog | level_n_pchainlog } ] [ -flag { pchainlog_chains_only | pchainlog_chains_and_failed_proc | pchainlog_complete } ] [ -flag { enable_pchainlog_bapi_msg | disable_pchainlog_bapi_msg } ] [ -flag { enable_pchain_details | disable_pchain_details } ] [ -flag { pchain_rerun | pchain_restart | pchain_refresh } ]
```

The parameters are described in Table 17: Task string parameters for SAP jobs on page 111.

Table 23. Task string parameters for SAP jobs

Parameter	Description	GUI Support
-job *job_name*	The name of the task to be run. It is either an InfoPackage technical field name, or a process chain name. This parameter is mandatory.	✓
-i {ipak_ | pchain_}	One of the following: **ipak_** Target job is an InfoPackage **pchain_** Target job is a process chain	✓
-debug	Turns on the most verbose `r3batch` trace. This option is for debugging the extended agent and should not be used in standard production.	✓
-trace	Turns on the `SAP RFC` trace. When you use this option, a trace file is created in the IBM Workload Scheduler `methods` directory. In UNIX, this trace file is called `dev_rfc`. In Windows, the file is called `rfc`*`XXXXX_XXXXX`*`.trc`. The `methods` directory is located in: UNIX **On UNIX operating systems** *`TWA_DATA_DIR`*`/methods` Windows **On Windows operating systems** *`TWA_home`*`\methods` This option is for debugging the extended agent and should not be used in standard production. Ensure that you delete the trace option from the job after you have	✓

Table 23. Task string parameters for SAP jobs (continued)

Parameter	Description	GUI Support
	performed debug procedures. The trace file can become very large and unmanageable.	
-flag {imm | immed}	Specifies to launch the job immediately, meaning that if there are no spare work processes, the job fails.	✓
-flag {enable_pchainlog | disable_pchainlog}	Enables or disables retrieval and appending of the process chain job log in the IBM Workload Scheduler `stdlist`. Disable if the size of the log affects performance. A related configuration option can be set for this purpose at a more general level. See Table 15: r3batch common configuration options on page 85.	✓
-flag {enable_ipaklog | disable_ipaklog}	Enables or disables retrieval and appending of the InfoPackage job log in the IBM Workload Scheduler `stdlist`. Disable if the size of the log affects performance. A related configuration option can be set for this purpose at a more general level. See Table 15: r3batch common configuration options on page 85.	
-flag {level_*n*_pchainlog | level_all_pchainlog}	Allows for retrieval of process chain logs down to the process chain level you specify. **level_*n*_pchainlog** Specifies that the process chains are logged down to, and including, the level represented by number *n*. **level_all_pchainlog** Specifies that all the process chains are logged. The default is `level_1_pchainlog`. A related configuration option can be set for this purpose at a more general level. See Table 15: r3batch common configuration options on page 85.	
-flag {pchainlog_chains_only | pchainlog_chains_and_failed_proc | pchainlog_complete}	Specifies what type of process chain-related logs will be retrieved.	

Table 23. Task string parameters for SAP jobs (continued)

Parameter	Description	GUI Support
	pchainlog_chains_only Only the process chains are logged. **pchainlog_chains_and_failed_proc** In addition to the process chains, all the processes that failed are also logged. **pchainlog_complete** The process chains and all processes are logged. The default is **pchainlog_complete**. A related configuration option can be set for this purpose at a more general level. See Table 15: r3batch common configuration options on page 85.	
-flag {enable_pchainlog_bapi_msg | disable_pchainlog_bapi_msg}	Enables or disables retrieval of additional messages from the BAPI calls from the SAP Business Warehouse process chains and appends them to the IBM Workload Scheduler `stdlist`.	
-flag {enable_pchain_details | disable_pchain_details}	Enables or disables the display of details about the process chain job. A related configuration option can be set for this purpose at a more general level. See Table 15: r3batch common configuration options on page 85.	✓
-flag {pchain_rerun | pchain_restart | pchain_refresh}	Determines the action that IBM Workload Scheduler performs when you rerun a job that submits a process chain. **pchain_rerun** IBM Workload Scheduler creates another process chain instance and submits it to be run again. **pchain_restart** IBM Workload Scheduler restarts the original process chain from the failing processes to the end.	✓

Table 23. Task string parameters for SAP jobs (continued)

Parameter	Description	GUI Support
	pchain_refresh IBM Workload Scheduler updates the status and details of the original process chain. For more details about rerunning a process chain, refer to Rerunning a process chain job on page 171.	

Note: Typically, the `-debug` and `-trace` options are for debugging the extended agent and should not be used in standard production.

The following is an example for an InfoPackage job whose technical field name is `ZPAK_3LZ3JRF29AJDQM65ZJBJF5OMY`:

```
-job ZPAK_3LZ3JRF29AJDQM65ZJBJF5OMY -i ipak_
```

Displaying details about Business Warehouse InfoPackages

About this task

To display details about a Business Warehouse InfoPackage, perform the following steps:

1. From the **Design** menu, click **Manage Jobs on SAP**.
2. In **Engine name**, select the name of the IBM Workload Scheduler engine connection from which you want to view SAP job details.
3. In **SAP Job Type**, select **Business Warehouse InfoPackage**.
4. In **Workstation name**, specify the workstation where the SAP job runs. If you do not know the object name, click the **...** (Browse) button. In the **Name** and **Location** panel, enter some characters of the object name (asterisk is supported as a wildcard) and click **Start**. From the displayed list, select the workstation you want to use, and click **OK**.
5. Click **Display**. The list of available jobs of type Business Warehouse InfoPackage for the specified engine is displayed.
6. Select the job for which you want to display the details and click **Details**.
7. When you have finished viewing the details for the job, click **OK** to return to the list of SAP jobs on the workstation specified.

Displaying details about a process chain job

You can view the details for a process chain job including any local subchains contained in the process chain.

Before you begin

Ensure you have performed the following steps before running this procedure:

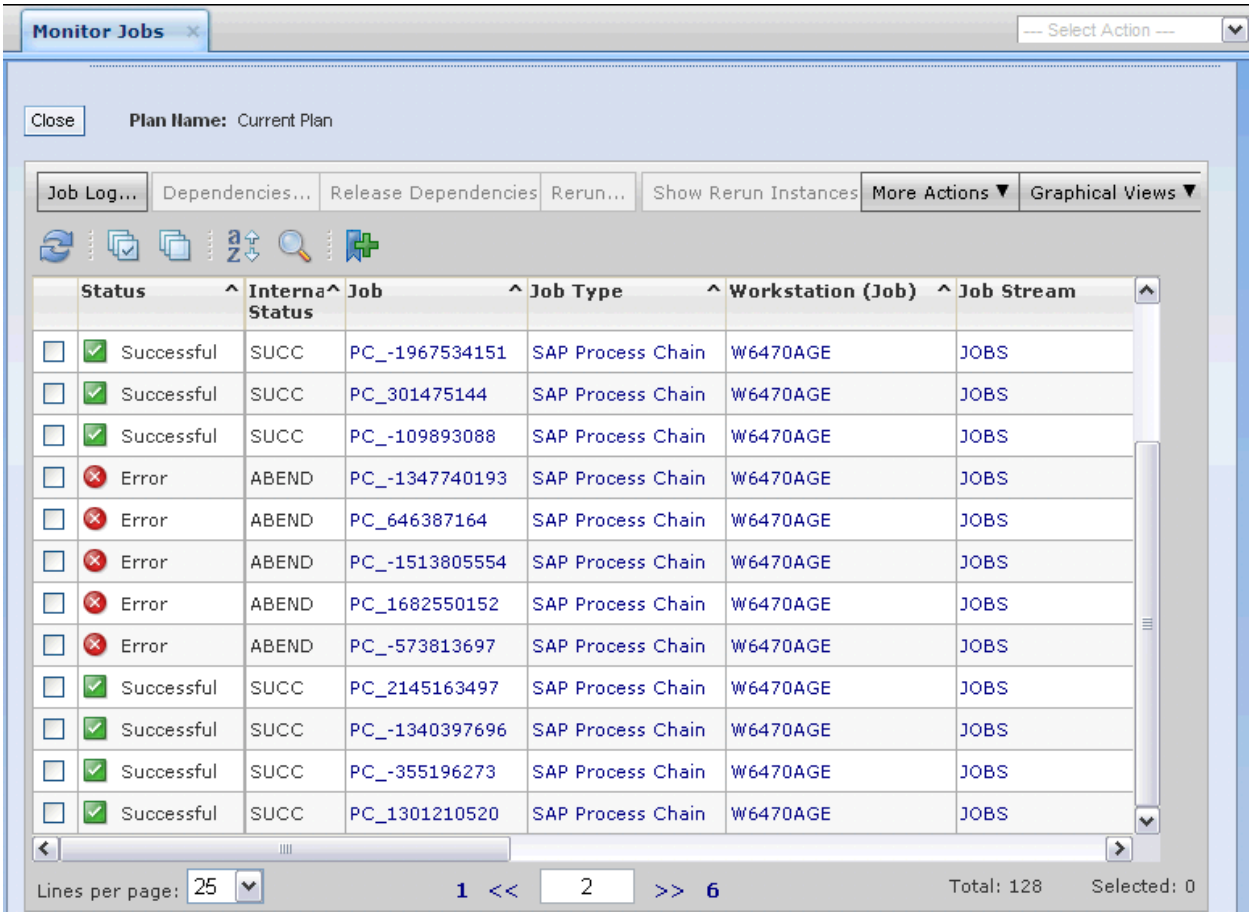
- Set the **pchain_details** option to `ON` in the common options file. For more information about this option, refer to Defining the common options on page 85.
- Distributed In a distributed environment, customize the Browse Jobs tasks that you created *before* installing IBM Workload Scheduler 8.4 Fix Pack 1 to show the **Job Type** column. For details about how to customize the task properties, refer to the Dynamic Workload Console online help.
- In a z/OS environment, you must customize the task properties to display the **Advanced Job Type** column that indicates the job type. For details about how to customize the task properties, refer to the Dynamic Workload Console online help.

About this task

To display details about an SAP Process Chain that you scheduled as an IBM Workload Scheduler job, perform the following steps from the Dynamic Workload Console.

1. Click **Monitoring and Reporting > Workload Monitoring > Monitor Workload**.
2. In **Engine name**, select the name of the IBM Workload Scheduler engine connection from where you want to work with SAP jobs.
3. In **Object Type**, leave the default selection **Job**.
4. Click **Run**.
5. The table of results corresponding to the search criteria is displayed:

Figure 12. Dynamic Workload Console - Table of results



6. Select a process chain job. For each process chain job, a hyperlink named **SAP Process Chain** is displayed.

Distributed

Distributed environment

The **Job Type** column displays **SAP Process Chain** to help you identify SAP process chain jobs.

z/OS

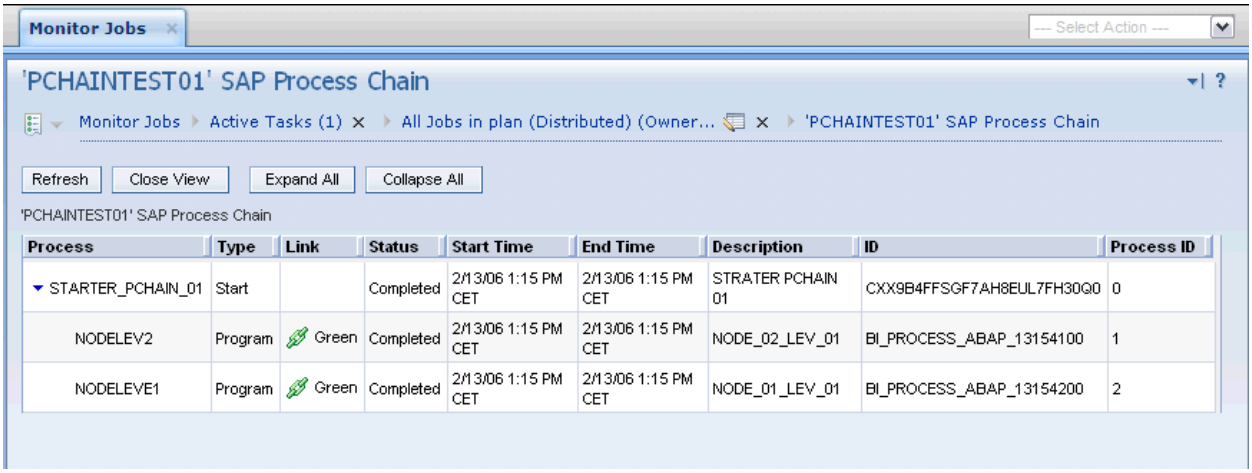
z/OS environment

The **Advanced Job Type** column displays **SAP Process Chain** to help you identify SAP process chain jobs.

Click the hyperlink for the job whose details you want to display.

7. The details for the process chain are displayed:

Figure 13. Dynamic Workload Console - Details of a process chain job



IBM Workload Scheduler monitors the process chain job until the job completes. The details shown reflect the last monitoring process performed. Perform a restart of the process chain indicating a refresh operation to synchronize the details with those on the remote SAP system to have the most updated information possible. If the process chain contains local subchains, a hyperlink is displayed for each one. Click the hyperlink you want, to display details about the corresponding subchain job. Alternatively, you can display the process chain details by clicking the hyperlink for the job and display the job properties panel. Click the hyperlink shown under **SAP Job Details**. The details for the process chain are displayed.

Rerunning a process chain job

Process chain jobs can be rerun from the start, rerunning the entire process chain, or they can be restarted from a specific process. Restarting a process enables you to restart without rerunning the whole process chain again. You can choose to either restart from the failed processes in a process chain, or restart a specific process indicating the related process ID.

To rerun an SAP job that submits a process chain, you can use one of the following user interfaces:

conman

For details, refer to the *IBM Workload Scheduler User's Guide and Reference*.

Dynamic Workload Console

See Procedure for rerunning a process chain job on page 175 for information about performing this task from the console.

For information about rerunning an SAP Standard R/3 job, see Rerunning a standard SAP job on page 126.

In general, when you rerun a process chain job, the new job is assigned the name of the alias you specify. To keep the original job name, set the IBM Workload Scheduler global option **enRetainNameOnRerunFrom** to `yes`. For details about this option, see *IBM Workload Scheduler Administration Guide*.

On extended agents, an alias is mandatory for each action you perform on the process chain job and the action itself, is the prefix of the alias name. For example, if you choose to restart a process chain from the failed processes, and assign PCHAIN1 as the alias for the process chain job, then the new job name is `Restart_PCHAIN1`.

z/OS In a z/OS environment, the process chain job maintains the same name and the Monitor Jobs view always displays the status for the last action performed on the job. Every time a rerun is performed on a process chain job, a new instance is generated each with a different ID.

Note:

1. By default, if you do not specify any setting, rerunning a process chain job corresponds to submitting a new process chain instance.
2. If you kill an IBM Workload Scheduler job that submits a process chain, the process chain is removed from schedule in the SAP Business Information Warehouse system. To restart the same process chain instance with r3batch, you require at least the following SAP Business Information Warehouse versions:
   - 3.0 with SP25
   - 3.1 with SP19
   - 3.5 with SP10
   - 7.0

   If your version of SAP Business Information Warehouse is earlier, you can restart the process chain only manually, through the SAP graphical interface.

   Table 24: Actions performed when you rerun a process chain job on page 172 shows the action performed when you rerun an IBM Workload Scheduler job that submits a process chain, depending on the settings you specify. These are the actions performed when you submit the rerun operation using the Rerun button from the Monitor Jobs view.

Table 24. Actions performed when you rerun a process chain job

Action performed	Description and setting
A new process chain instance is submitted	IBM Workload Scheduler creates another process chain instance and submits it to be run again. This action occurs when: • On extended agents, you specify `RERUN`*`value`* as the step to rerun, where *`value`* is any value you want. This setting overrides the settings in the job definition and options file, if any. In an end-to-end environment, you can perform this action on a centralized job by adding the following parameter to the script file: `-flag pchain_rerun`

Table 24. Actions performed when you rerun a process chain job (continued)

Action performed	Description and setting
	• In the job definition, you set `-flag pchain_rerun`. This setting overrides the setting in the options file, if any. For a description of this parameter, see Table 23: Task string parameters for SAP jobs on page 165. • In the options file, you set the **pchain_recover** option to `rerun`. For a description of this option, refer to Table 15: r3batch common configuration options on page 85.
The original process chain is rerun from the failed processes	IBM Workload Scheduler restarts the original process chain from the failed processes to the end. In this way, after you detected the error that caused the failure and performed the recovery action, you can rerun the process chain job from the failed processes and have its run completed. This action is performed only if at least one process in the process chain did not complete successfully. It occurs when: • On extended agents, you specify `RESTART`*`value`* as the step to rerun, where *`value`* is any value you want. This setting overrides the settings in the job definition and options file, if any. In an end-to-end environment, you can perform this action on a centralized job by adding the following parameter to the script file: `-flag pchain_restart` • In the job definition, you set `-flag pchain_restart`. This setting overrides the setting in the options file, if any. For a description of this parameter, see Table 23: Task string parameters for SAP jobs on page 165. • In the options file, you set the **pchain_recover** option to `restart`. For a description of this option, refer to Table 15: r3batch common configuration options on page 85.
The process that you specify is restarted	IBM Workload Scheduler restarts the process of the original process chain that you specify, and monitors the process chain run until its final state. On extended agents, this action occurs when you specify `PROCESS`*`processID`* as the step to rerun, where *`processID`* is the identifier of the process you want. For example, if the process ID is `3`, you must specify `PROCESS3` as the step. You can view the process IDs in the following ways:

Table 24. Actions performed when you rerun a process chain job (continued)

Action performed	Description and setting

	• Dynamic Workload Console, version 8.5 or later. From the panel where the details about the process chain are displayed, see the column named Process ID. For details about how to display the process chain details, refer to Displaying details about a process chain job on page 168.
	• IBM Workload Scheduler job log, as follows:
	`+++ EEWO1071I Start of process chain PCHAIN1`
	`Process Chain PCHAIN1 (Log ID:D3C0ZWAYESD58PXOYPEOGNZK7).`
	`-------------------------`
	`Process Type: TRIGGER.`
	`Process Variant: PCHAIN1_STARTER.`
	`Actual State: F.`
	`...`
	`>> **Process ID: 3.**`
	`Process Type: ABAP.`
	`Process Variant: Z_PCHAIN1_NODE3.`
	`Actual State: F.`
	`Instance: D3C0ZXL3IJ8LR5O9Q1D9A4Y4N.`
	`>> **Process ID: 4.**`
	`Process Type: ABAP.`
	`Process Variant: Z_PCHAIN1_NODE1.`
	`Actual State: .`
	`Instance: D3C0ZZKS0RR88DKRJQ09Z1WW7.`
	`+++ EEWO1072I End of process chain PCHAIN1`
	The following list shows the meaning of the alphabetic value used as the actual state in the job log:
	Actual state
	Meaning
	A
	Active
	F
	Completed
	G
	Successfully completed
	P
	Planned
	Q
	Released

Table 24. Actions performed when you rerun a process chain job (continued)

Action performed	Description and setting
	R Ended with errors **S** Skipped **X** Canceled **Y** Ready ***blank*** Undefined In an end-to-end environment, you can perform this action on a centralized job by adding the following parameter to the script file: `-pchain_pid` *processID*
The status and details of the original process chain are updated	IBM Workload Scheduler monitors the original process chain until its final status. This action occurs when: • On extended agents, you specify `REFRESH`*`value`* as the step to rerun, where *`value`* is any value you want. This setting overrides the setting in the job definition, if any. In an end-to-end environment, you can perform this action on a centralized job by adding the following parameter to the script file: `-flag pchain_refresh` • In the job definition, you set `-flag pchain_refresh`. For a description of this parameter, see Table 23: Task string parameters for SAP jobs on page 165.

Procedure for rerunning a process chain job

You can rerun all of the processes in the process chain from the Dynamic Workload Console or you can rerun at a process level.

Before you begin

z/OS In z/OS environments, you need to set the status of the job to **Ready** before you can rerun the job.

1. Select a job and click **Set Status**.
2. In **Change Status**, select **Ready**.
3. Click **OK** to return to the list of jobs.

About this task

To rerun a process chain SAP job, perform the following steps:

1. Click **Monitoring and Reporting > Workload Monitoring > Monitor Workload**.
2. In **Engine name**, select the name of the IBM Workload Scheduler engine connection from where you want to work with SAP jobs.
3. In **Object Type**, leave the default selection **Job**.
4. Click **Run**.
5. A list of jobs is displayed. Select a process chain job.

   Distributed

   Distributed **Distributed environment**

   The **Job Type** column displays **SAP Process Chain** to help you identify SAP process chain jobs.

   z/OS

   z/OS environment

   The **Advanced Job Type** column displays **SAP Process Chain** to help you identify SAP process chain jobs. To display the **Advanced Job Type** column in the table, edit the **Task Properties** and in **Column Definition**, add the **Advanced Job Type** column to the **Selected Columns** list. Move the column up to define the order of the column in the table and make it more visible.
6. Rerun the job.

   a. Click **More Actions > Restart Process Chain**.

   b. Select the action you want to perform on the selected process chain:

   Rerun

   Reruns the entire process chain. The process chain ID on the SAP system remains the same, as well as the job identifier on z/OS systems.

   Distributed Specify an alias to identify the new job. In distributed systems the rerun process chain is identified with this alias name prefixed by `RERUN`.

   Refresh

   Refreshes the Dynamic Workload Console view with the latest updates on the remote SAP system so that the two views are synchronized.

   Distributed Specify an alias to identify the new job. In distributed systems the refreshed process chain is identified with this alias name prefixed by `REFRESH`.

Restart from the failed processes

Action available only for process chains in error state. Rerun only some steps of the process chain, starting from the failed processes.

Distributed Specify an alias to identify the new job. In distributed systems the restarted process chain is identified with this alias name prefixed by `RESTART`.

Restart from a specific process

Action available only for process chains in error state. Rerun only some steps of the process chain, starting from the process specified in the **SAP Process ID** field. You can find the process ID by opening the job log or viewing the job type details from the table of results of your monitor job task.

Distributed In distributed systems the restarted process chain is identified with this alias prefixed by `PROCESS`.

7. Click **OK** to perform the selected action on the process chain.

Results

The job reruns immediately.

Business scenario: rerunning the original process chain job from the failed process

As a scheduling administrator, you are responsible for managing batch jobs in both SAP and non-SAP systems. The workflow is one or more job streams in IBM Workload Scheduler. A job stream contains jobs that collect and prepare data for month-end closing over all sales channels. The month-end closing report requires data to be collected from several sales and distribution systems. Data is collected using local and remote process chains in the SAP Business Intelligence system. The process chains include a set of Infopackages, ABAP reports, and operating system jobs to sort the report data by a logical hierarchy.

To administer from a single point of control, you link the SAP process chains to IBM Workload Scheduler through IBM Workload Scheduler.

During batch processing, an IBM Workload Scheduler job comprising a process chain, failed. Optionally, you can see which processes failed either from the Dynamic Workload Console (for details, see Displaying details about a process chain job on page 168) or in the job log. You ask the SAP administrator to fix the cause of the error, then, on an extended agent, you rerun the IBM Workload Scheduler job by setting the step as `RESTART`*`value`*. In this way, the original process chain is restarted from the failed processes and continues until the ending step.

Alternatively, you can select the process chain job from the Monitor Jobs view on the Dynamic Workload Console and then select **More Actions > Restart Process Chain** and then select the **Restart from the failed processes** option.

Business scenario: restarting a specific process of the process chain

You might decide to restart a single process as a preparation step before restarting the failed processes of a process chain. A failed process might have corrupted some data, so you run the single process to restore the data and set up the required system state before you rerun the other processes in the process chain.

Suppose you are using InfoPackages and process chains to extract data from one or several sources and you want to transform this data into managerial reports, for example by using aggregate functions. If the process that transforms this data fails, it might corrupt the data that the preceding InfoPacakge process had successfully extracted. After fixing the problem with the transformation process, you must restart the InfoPackage extraction process to reload the data, even though this extraction process had completed successfully before. Restart the failed transformation process only after the data has been reloaded, either by restarting the failed processes of the process chain or by restarting just the failed transformation process.

On an extended agent, from the Monitor Jobs view on the Dynamic Workload Console, select the process chain and click Rerun, then specify `PROCESS`*`processID`* as the step to rerun, where *processID* is the identifier of the process you want to restart.

To restart a specific process of the process chain, from the Monitor Jobs view on the Dynamic Workload Console, select the process chain and click **More Actions > Restart Process Chain** and then select the **Restart from a specific process** option, specifying the process ID in the **SAP Process ID** field.

Job throttling feature

Learn how the job throttling feature helps you to improve the efficiency of your scheduling on SAP systems and reduce the batch window for your SAP jobs to a minimum.

Using advanced XBP 2.0 and 3.0 functions, such as the job interception and parent-child, the job throttler ensures that the SAP system is not overloaded and the number of released jobs does not exceed the total number of SAP background work processes in the system.

You can also configure the job throttler to send data related to its activity to the SAP Computing Center Monitoring System (CCMS) for monitoring purposes.

Business scenario

You manage your Internet sales through an application software that verifies that data is correct, checks the availability of the item, and validates the order. To process all the orders received, you scheduled an IBM Workload Scheduler job to run every 12 hours, connect to SAP, and generate a child job for every order to process. Child jobs are in charge of creating shipping bills, checking destination address, and forwarding the orders to the appropriate carrier, thus optimizing the delivery process. A potential overload of the system might occur during peak times, for example over Christmas, and could risk the late delivery of orders, damaging your business. To manage the submission of jobs and activate an advanced management of their priority class (for both parent and child jobs), enable the job throttling feature.

Additionally, you might want to set a policy so that an SAP CCMS alert is raised each time the number of jobs to be released under the control of the job throttler exceeds a certain threshold. To do this, you enable the job throttler to send data to the SAP CCMS monitoring architecture. At job throttler startup, an MTE that monitors the number of jobs to be released by the

job throttler is created. By including the MTE in a monitoring set and specifying the related threshold, you are alerted each time the threshold is exceeded.

Software prerequisites

To use job throttling, you must have the SAP JCo 3.0.2 libraries or later (`dll` and `jar` files) installed in the `<data_dir>/methods/throttling/lib` directory. To download JCo 3.0.x, visit the SAP Service Marketplace web site.

Setting and using job throttling

The job throttler enqueues intercepted jobs and releases them when the background work processes that they need on the SAP server or SAP server group are available. The queue of intercepted jobs is sorted by scheduling time and priority of SAP jobs. When the SAP parent-child feature is enabled, child jobs inherit their progenitor's priority so that new urgent jobs are run before other planned jobs.

The following sections describe the steps to operate job throttling.

Step 1. Setting the options in the options file

About this task

To define the behavior of the job throttling feature, set the following options in the options file. For detailed information about the options, see Table 15: r3batch common configuration options on page 85.

- `throttling_enable_job_class_inheritance`
- `throttling_enable_job_interception`
- `throttling_interval`
- `throttling_max_connections`
- `throttling_release_all_on_exit`

Step 2. Enabling and configuring the job interception feature

About this task

As a prerequisite, the job throttler requires that the job interception feature is enabled in the SAP system. To enable and configure job interception, follow these steps.

Note: Ensure that the job throttling and job interception features are not running at the same time. The job throttler cannot start if interception collector jobs are running.

1. Enable job interception, either automatically or manually, as follows:

   Automatic activation (meaning that the job throttler enables the job interception on SAP system)

   In the options file of the workstation with the r3batch access method you are using, set `throttling_enable_job_interception=on` (this is the default).

Manual activation

a. In the SAP system, run the INITXBP2 ABAP program in the transaction `se38` and enable job interception.
b. In the options file of the workstation with the r3batch access method you are using, set `throttling_enable_job_interception=off`.

Note: When you stop the job throttler, the setting for the job interception feature that was previously configured on the SAP system is restored.

2. In the SAP system, configure the job interception criteria as follows:
   a. Launch the transaction `se16` to access the table TBCICPT1, where the interception settings are maintained.
   b. Set the job name, creator, and client related to the jobs you want to intercept. To intercept all SAP jobs, specify the wildcard * (asterisk) for the job name, creator, and client.
   c. Save your settings and close the dialog.

SAP will intercept all the jobs matching the selection criteria, and the job throttling will release all the jobs that were intercepted.

Step 3. Enabling job class inheritance

About this task

You can configure the job throttler to have the intercepted job inherit the priority class from its progenitor (the top-level job in the hierarchy), if the progenitor class is higher than the intercepted job class. To do this, in the options file set `throttling_enable_job_class_inheritance=on`; this setting automatically enables the parent-child feature on the SAP system.

Note: When you stop the job throttler, the setting for the parent-child feature that was previously configured on the SAP system is restored.

Step 4. Configuring the logging properties

About this task

You can configure the trace properties of the job throttler by editing the logging configuration file `jobthrottling.properties` located in `<data_dir>/methods/throttling/properties`.

To configure the trace level, follow the procedure.

1. Set the trace level property. The supported trace levels are: `DEBUG_MIN`, `DEBUG_MID`, and`DEBUG_MAX`, where `DEBUG_MAX` is the most verbose trace level.
2. Save the changes.

Results

When making changes to the trace level setting, the changes are effective immediately after saving the .properties file. Other changes might require a restart to make them effective.

What to do next

You can also configure the name, number, and size of the trace file. By default, the job throttler generates a maximum of 3 files of 5 MB in the `<data_dir>/methods/traces` directory.

Note: The job throttler creates the `<data_dir>/methods/traces` directory as soon as it is started.

Step 5. Starting and stopping the job throttling feature

About this task

To start job throttling, run the `jobthrottling` executable file related to the operating system you are using. Optionally, you can create an IBM Workload Scheduler job that starts the job throttler.

Note: On Windows systems using a single-byte character language, to start job throttling from a command prompt ensure that the DOS shell font is *not* Lucida Console. Ensure also that you set the IBM Workload Scheduler environment by entering the following command:

```
TWA_home\tws_env.cmd
```

From a command prompt, enter:

UNIX operating systems

```
TWA_home/methods/jobthrottling.sh {XA_Unique_ID|base_options_filename} [-scratch]
```

Windows operating systems

```
TWA_home\methods\jobthrottling.bat {XA_Unique_ID|base_options_filename} [-scratch]
```

Where:

XA_Unique_ID

The unique identifier for the extended agent workstation you are using. See UNIQUE_ID on page 21 for details about retrieving the unique identifier for a workstation.

base_options_filename

For dynamic and z-centric agents, the file name of the options file without the extension, defined on the engine workstation hosting the workstation with the r3batch access method.

-scratch

If you enabled the job throttler to send data to CCMS (for details, see Sending data from job throttling to the CCMS Monitoring Architecture on page 182), the job throttler starts and resets the attribute MTE named `JT total released jobs` to 0. If you do not specify **-scratch**, the job throttler starts and increments the `JT total released jobs`.

This parameter is optional, and has effect only if the job throttler sent its data to CCMS at least once before.

To know the syntax for the `jobthrottling` command, run the command as follows:

```
jobthrottling -u
```

To stop the job throttler, enter the following command (optionally, you can create an IBM Workload Scheduler job that stops the job throttler):

UNIX operating systems

TWA_home/methods/stop-jobthrottling.sh {*XA_Unique_ID*|*base_options_filename*}

Windows operating systems

TWA_home\methods\stop-jobthrottling.bat {*XA_Unique_ID*|*base_options_filename*}

Alternatively, you can enter the following command (you must be connected as *TWSUser* and have read and write permissions on the `txt` file):

```
echo shutdown > TWA_home/methods/{XA_Unique_ID|base_options_filename}_jobthrottling_cmd.txt
```

The job throttler stops:

- When the timestamp of {*XA_Unique_ID*|*base_options_filename*}_jobthrottling_cmd.txt is later than the time when the job throttler started.
- Within the time interval you specified in the `throttling_interval` option.

Sending data from job throttling to the CCMS Monitoring Architecture

About this task

You can configure the job throttler to send data related to its activity to the SAP Computing Center Monitoring System (CCMS) for monitoring purposes. Sending data from the job throttler to CCMS is supported if you have at least the SAP Web Application Server 6.20, Support Package 12 installed.

In the options file, set the following options (for details, see Table 15: r3batch common configuration options on page 85):

```
throttling_send_ccms_data
throttling_send_ccms_rate
```

In this way, at job throttler startup the following monitoring tree elements (MTE) are created:

- A context MTE named **ITWS for Apps**.
- An object MTE with the same name as the IBM Workload Scheduler extended agent where the job throttler is running. This object MTE belongs to the context MTE **ITWS for Apps**.
- The following attribute MTEs:

JT total released jobs

The total number of jobs that the job throttler has released since startup. This value depends on the `-scratch` option you set at job throttler startup; for details, see Step 5. Starting and stopping the job throttling feature on page 181.

JT queue

The number of enqueued intercepted jobs to be released.

JT released jobs per cycle

The number of released jobs in the latest run. This value depends on the `throttling_send_ccms_rate` setting; for details, see Table 15: r3batch common configuration options on page 85.

Note: By default `throttling_release_all_on_exit` is set to `ON`, meaning that when you stop the job throttler, all the intercepted jobs are released. However, these jobs are not considered when updating the **JT total released jobs**, **JT queue**, and **JT released jobs per cycle** MTEs.

To begin monitoring, include the MTEs in the monitoring set you want, and set the thresholds to generate an alert.

You can define an IBM Workload Scheduler event rule based on the CCMS alerts; for detailed information, refer to Defining event rules based on CCMS Monitoring Architecture alerts on page 206.

For example, to define an event that monitors the attribute MTE **JT total released jobs**, on the extended agent workstation with unique identifier `SAP_XA`, connected to the SAP system ID `T01`, specify the following information:

XA Workstation

`SAP_XA`

MTE SAP System ID

`T01`

MTE Monitoring Context Name

`ITWS for Apps`

MTE Monitoring Object Name

`SAP_XA`

MTE Monitoring Attribute Name:

`JT total released jobs`

Deleting the monitoring tree elements

About this task

After you stopped the job throttling feature, if you configured it to send its status data to CCMS, you can delete one or more MTEs that were created. To do this:

1. From the SAP GUI, invoke the transaction `rz20` to display a list of monitor sets.
2. Locate the monitor set named SAP CCMS Technical Expert Monitors, and expand it.
3. Locate the monitor named All Monitor Contexts, and double-click it to open it.
4. From the action menu, select **Extras -> Activate Maintenance Functions**.
5. Locate the MTE named **ITWS for Apps** and select it.
6. Right-click the MTE and select **Delete**. You are prompted to choose one of the delete options.
7. Select the option you want. The MTE is deleted accordingly.

Note: Deleting **ITWS for Apps** from the All Monitor Contexts monitor, deletes also all the copies that you might have created in other monitors.

Exporting SAP factory calendars

This section describes how to export SAP factory calendars into a file format that can be processed by the IBM Workload Scheduler **composer** command line, to add the exported calendar definitions to the IBM Workload Scheduler database.

Business scenario

About this task

You might want to configure your IBM Workload Scheduler scheduling activities based on the schedule calendar in your SAP system. To do this, use the `r3batch` export function to export the SAP calendar definitions into a file whose format is compatible with the IBM Workload Scheduler **composer** command line. Based on the parameters you specify, you create a file that contains only the SAP calendar definitions that meet your scheduling requirements. Use this file as input for the **composer add** command, to import the calendar definitions into the IBM Workload Scheduler database. Your IBM Workload Scheduler and SAP calendars are now synchronized.

To keep the IBM Workload Scheduler and SAP calendar definitions synchronized and avoid duplicating data maintenance in the two environments, you can schedule to export the calendar definitions from SAP and import them to IBM Workload Scheduler on a regular basis using a dedicated job.

Exporting and importing SAP factory calendars

Refer to the following sections:

- Exporting factory calendars on page 185 for an explanation about how you use the **r3batch** export function to access and download factory calendars available in an SAP system. The main purpose of this function is to create an output file that can be used by the **composer** to synchronize IBM Workload Scheduler calendars with existing SAP factory calendars, integrating the calendar definitions from SAP into IBM Workload Scheduler.
- Importing factory calendars on page 187 for an explanation about how you import the exported calendar definitions into the IBM Workload Scheduler database.

For details about the IBM Workload Scheduler calendar definitions, see *User's Guide and Reference*.

Exporting factory calendars

About this task

To export an SAP calendar, from `TWA_home/methods` (where *TWA_home* is the complete path where you installed IBM Workload Scheduler) enter the following command:

Command syntax

```
r3batch -t RSC -c XA_Unique_ID -- " -calendar_ID calendarID -year_from yyyy -year_to yyyy [ { -getworkdays | -getfreedays } ] [ -tws_name tws_cal_name ] [ -tws_description tws_cal_desc ] [ -filename output_filename ] "
```

Where:

-t RSC

The identifier of the task to be performed, in this case RSC (Retrieve SAP Calendars). This parameter is required.

-c *XA_Unique_ID*

The unique identifier for the extended agent workstation connected to the SAP system where the calendar data to export is located. The SAP system must be configured as a workstation to IBM Workload Scheduler. This parameter is required. For information about retrieving the unique identifier for the extended agent workstation, see UNIQUE_ID on page 21.

-calendar_id *calendarID*

The identifier of the SAP calendar to be exported, which consists of two alphanumeric characters. This parameter is required.

-year_from *yyyy*

The year of the calendar from when to start exporting dates, in the format *yyyy*. This parameter is required.

-year_to *yyyy*

The year of the calendar when to stop exporting dates, in the format *yyyy*. This parameter is required.

-getworkdays | -getfreedays

Specify getworkdays to create the IBM Workload Scheduler calendar definition based on the working days of the SAP calendar. In this way, each date of a working day is stored in the output file.

Specify getfreedays to create the IBM Workload Scheduler calendar definition based on the holidays of the SAP calendar. Each date of a non-working day is stored in the output file.

These parameters are optional and mutually exclusive. If you do not specify either, the default is getworkdays.

-tws_name *tws_cal_name*

The IBM Workload Scheduler name for the exported SAP factory calendar. It is stored in the output file.

You can specify up to eight alphanumeric characters. This parameter is optional, the default is SAP*XX_calendarID*, where:

XX

Corresponds to WK if the calendar includes only working days or FR if the calendar includes only non-working days.

calendarID

The identifier of the SAP calendar.

For example, the default IBM Workload Scheduler name for an exported calendar, whose identifier is 04, that includes only working days, is SAPWK_04.

-tws_description *tws_cal_desc*

The description of the IBM Workload Scheduler calendar. It is stored in the output file. You can specify up to 120 alphanumeric characters. If the description contains blanks, it must be enclosed between single quotes. This parameter is optional.

-filename *output_filename*

The name of the output file that is to contain the calendar definitions. This file is written in a scheduling language that can be processed by the **composer** when you add the calendar data to the IBM Workload Scheduler database.

You can specify a file name with its complete or partial path; if you do not specify any path, the file is created in the current directory. If the path you specify does not exist, it is created, provided that you have the appropriate access rights. Otherwise, the command returns an error message and is not performed.

You can specify up to the maximum number of characters allowed by your operating system. If the name of the file contains blanks, it must be enclosed between single quotes. If another file with the same name exists, it is overwritten.

This parameter is optional. The default value is *tws_name*.txt, where *tws_name* is the value you set for the tws_name parameter.

The following is an example of an SAP factory calendar export command:

```
r3batch -t RSC -c horse10 -- " -calendar_id 01 -year_from 2007
-year_to 2010 -tws_name CAL1 -tws_description 'SAP Calendar 01'
-getworkdays -filename 'my dir/calendar_01.dat' "
```

This command exports the SAP calendar named 01, located on the SAP system named horse10. The dates exported begin from year 2007, until year 2010, considering only working days. The IBM Workload Scheduler name used for the calendar is CAL1, and the description written in the output file is SAP Calendar 01. The output file is named `calendar_01.dat`, stored in `<data_dir>/methods/my dir`, and its content looks like the following

```
$CALENDAR
    CAL1
    "SAP Calendar 01"
    01/02/2007 01/03/2007 01/04/2007 01/05/2007 01/08/2007 01/09/2007 01/10/2007
    01/11/2007 01/12/2007 01/15/2007 01/16/2007 01/17/2007 01/18/2007 01/19/2007
    01/22/2007 01/23/2007 01/24/2007 01/25/2007 01/26/2007 01/29/2007 01/30/2007
```

```
    01/31/2007 02/01/2007 02/02/2007 02/05/2007 02/06/2007 02/07/2007 02/08/2007
     .......
    11/24/2010 11/25/2010 11/26/2010 11/29/2010 11/30/2010 12/01/2010 12/02/2010
    12/03/2010 12/06/2010 12/07/2010 12/08/2010 12/09/2010 12/10/2010 12/13/2010
    12/14/2010 12/15/2010 12/16/2010 12/17/2010 12/20/2010 12/21/2010 12/22/2010
    12/23/2010 12/24/2010 12/27/2010 12/28/2010 12/29/2010 12/30/2010 12/31/2010
```

Importing factory calendars

About this task

To import the exported calendar definitions into the IBM Workload Scheduler database, copy the output file from the extended agent for SAP to the master workstation and from the **composer** command line on the master workstation, enter the following command:

```
-add output_filename
```

where `output_filename` is the name of the exported file, with its complete path.

For example, to import the `tws_calendar_01.dat` file exported in the previous example, copy the file to the master workstation. From the **composer** command line on the master workstation, enter:

```
-add TWA_home/methods/my dir/tws_calendar_01.dat
```

where *TWA_home* is the complete path where you installed IBM Workload Scheduler.

Defining internetwork dependencies and event rules based on SAP background events

This section describes how to define internetwork dependencies and event rules for IBM Workload Scheduler based on SAP background events.

Note: To be able to define and monitor event rules, you must configure your environment as described in Configuring SAP event monitoring on page 100.

Defining internetwork dependencies based on SAP background events

Dependencies are prerequisites that must be satisfied before a job or job stream can start. Internetwork dependencies are dependencies checked by the extended agent workstation to which they belong. In response to an internetwork dependency, the SAP extended agent checks for the occurrence of the SAP background event specified in the dependency. As soon as the SAP event is raised, the SAP extended agent commits the event and instructs IBM Workload Scheduler to resolve the corresponding internetwork dependency.

For more details about internetwork dependencies, refer to the *IBM Workload Scheduler: User's Guide and Reference*. For more details about how to raise SAP events, see Raising an SAP event on page 125.

To define SAP background events as internetwork dependencies, XBP versions 2.0 and 3.0 are supported, with the following differences:

XBP version 2.0

SAP background events can release IBM Workload Scheduler internetwork dependencies only if the dependencies are created or checked *before* the SAP event is raised. An event history is ignored, therefore an SAP event raised before the internetwork dependency is created, is not considered.

Note: Because an SAP event history is ignored, for each SAP background event to be checked, a placeholder SAP job is created. This is a dummy job whose running depends on the SAP background event, therefore an SAP event is considered raised as soon as the corresponding placeholder job has completed.

XBP version 3.0 (supported by SAP NetWeaver 7.0 with SP 9, or later)

Only the SAP background events stored in the SAP event history table are considered by IBM Workload Scheduler to check for internetwork dependencies resolution. As a prerequisite, the SAP administrator must create the appropriate event history profiles and criteria on the target SAP system.

To avoid performance reduction, run reorganization tasks against the SAP event history.

Note: Some SAP systems providing XBP version 3.0 still return XBP version as 2.0. To check if your SAP system provides XBP 3.0, invoke the transaction `se37` and search for the function module `BAPI_XBP_BTC_EVTHISTORY_GET`. If your system contains the module, set the `xbpversion` option to 3. In this way, **r3batch** will ignore the XBP value returned by the SAP system. For details about the `xbpversion` option, refer to Table 15: r3batch common configuration options on page 85.

To define an SAP background event as an internetwork dependency, use the following parameters:

Table 25. Parameters to define an SAP internetwork dependency

Parameter	Description	GUI support
-evtid *sap_event_name*	The name of the SAP background event, up to 32 characters. If the name contains blanks, enclose it between single quotes. This parameter is required.	✓
-evtpar *sap_event_parm*	The SAP event parameter, up to 64 characters. If the parameter contains blanks, enclose it between single quotes. This parameter is optional.	✓
-commit	Defines that the SAP background event is committed immediately after the internetwork dependency has been resolved. If you do not specify `-commit`, the event must be committed by running the `r3batch` task PI. The default is that `-commit` is not specified. For details about the PI task, refer to Committing SAP background events by an external task on page 190. In addition to this parameter, you can set as default that the system commits internetwork dependencies immediately by specifying `commit_dependency=on` in	✓

Table 25. Parameters to define an SAP internetwork dependency (continued)

Parameter	Description	GUI support
	the options file. For details about the commit_dependency option, see Table 15: r3batch common configuration options on page 85. **Note:** With XBP version 2.0, defining two internetwork dependencies on the same SAP event might lead to an error, if `-commit` is specified. For example, suppose you define an internetwork dependency for the SAP event `SAPEVT`, with or without setting `-commit`. After this definition, the SAP event `SAPEVT` is raised. Then you define a second internetwork dependency based on `SAPEVT`, specifying `-commit`. The second dependency immediately commits the SAP event, with the consequence that the first dependency becomes impossible to resolve. Therefore, when the first job checks for the internetwork dependency, an error is issued.	

The following example shows how to define an internetwork dependency based on the SAP background event named `SAP_TEST` with the parameter `12345678`. After its processing, the event is not immediately committed.

```
-evtid SAP_TEST -evtpar 12345678
```

The resulting internetwork dependency looks like the following, where `SAPWS` is the name of the extended agent workstation that connects to the SAP background processing system where the event runs:

```
follows SAPWS::"-evtid SAP_TEST -evtpar 12345678"
```

The following example shows how to define an internetwork dependency based on the SAP background event named `SAP_TEST`, without parameter. As soon as the internetwork dependency is resolved, the event is committed.

```
-evtid SAP_TEST -commit
```

The resulting internetwork dependency looks like the following, where `SAPWS` is the name of the extended agent workstation that connects to the SAP background processing system where the event runs:

```
follows SAPWS::"-evtid SAP_TEST -evtpar 12345678"
```

Table 26: Internetwork dependency definition and possible resolution on page 189 shows the correspondence between the definition and possible resolution of an internetwork dependency that depends on an SAP event, with or without parameters assigned. In this table, `SAP_TEST` is used as the event name and `12345678` or `ABCDEFG` as the event parameter.

Table 26. Internetwork dependency definition and possible resolution

IBM Workload Scheduler internetwork dependency specified	SAP event raised in SAP system	SAP event pa rameter	IBM Workload Scheduler internetwork dependency resolved
-evtid SAP_TEST	none	none	No

Table 26. Internetwork dependency definition and possible resolution (continued)

IBM Workload Scheduler internetwork dependency specified	SAP event raised in SAP system	SAP event parameter	IBM Workload Scheduler internetwork dependency resolved
-evtid SAP_TEST	END_OF_JOB	none	No
-evtid SAP_TEST	SAP_TEST	none	Yes
-evtid SAP_TEST	SAP_TEST	12345678	Yes
-evtid SAP_TEST -evtpar 12345678	SAP_TEST	none	No
-evtid SAP_TEST -evtpar 12345678	SAP_TEST	12345678	Yes
-evtid SAP_TEST -evtpar 12345678	SAP_TEST	ABCDEFG	No

Committing SAP background events by an external task

About this task

SAP events defined as IBM Workload Scheduler internetwork dependencies, by default are not automatically committed after their processing. You can modify this default by specifying the `-commit` parameter. Otherwise, if you leave the default, you must commit the processed event by using the external task Put Information (PI).

The PI task commits all the processed events that meet the given criteria. For this reason, it is recommended that you run this task at the end of the working day. By doing so, internetwork dependencies that are already resolved are not reset and the objects depending on them are not blocked until they are resolved again.

From a command line, enter the following command:

Command syntax

-r3batch -t PI -c *XA_Unique_ID* **-- " -t CE -evtid***sap_event_name* [**-evtpar***sap_event_parm*] **"**

Where:

-t PI

The identifier of the task to be performed, in this case PI (Put Information). This parameter is required.

-c *XA_Unique_ID*

The unique identifier for the extended agent workstation connected to the SAP background processing system where the event is run. This parameter is required. For information about retrieving the unique identifier for the extended agent workstation, see UNIQUE_ID on page 21.

-t CE

The identifier of the task to be performed, in this case CE (Commit Event). This parameter is required.

-evtid ***sap_event_name***

The name of the SAP event running on the background processing system. If the name contains blanks, enclose it between single quotes. This parameter is required.

-evtpar ***sap_event_parm***

The parameter of the SAP event running on the background processing system. If the parameter contains blanks, enclose it between single quotes. This parameter is optional. If you do not specify it, all the SAP events with the name you specified, with or without a parameter, are committed on the target system.

The following is an example of how to commit the SAP event named `SAP_TEST`, with parameter `1234567`, with extended agent workstation with unique identifier `horse10` connected to the background processing system:

```
r3batch -t PI -c horse10 -- " -t CE -evtid SAP_TEST -evtpar 1234567"
```

Defining internetwork dependencies based on SAP background events with the Dynamic Workload Console

About this task

To define an SAP background event as an internetwork dependency with the Dynamic Workload Console, perform the following steps:

1. Launch the **Workload Designer** from the Dynamic Workload Console. From the **Design** menu, click **Workload Designer** page.
2. Search for and open the job stream you want to manage.
    a. Select the **Job stream card** to display all job streams present in the selected folder. Otherwise, you can select the **Job stream card** and then type the job stream name in the **Search** bar.
    b. Select the job stream and click **Edit**. The job stream and its contents are displayed in the **Workspace area**.
3. In the job stream subrow, click **Add Dependency** and select **Internetwork**.
4. Specify the properties for the internetwork dependency.
    a. In the **Network Agent** field, enter the name of the agent workstation connected to the SAP background processing system where the event runs.
    b. In the **Dependency** field, enter the parameters to define the internetwork dependency. For a description of the parameters allowed, refer to Table 25: Parameters to define an SAP internetwork dependency on page 188.
5. Click **Save** to save the changes to the job stream.

Results

The local job or job stream now has a dependency on a SAP background event. You can also perform this procedure from the graphical view available from the job stream row in the **Workspace area**. For more information about adding dependencies and editing objects in the graphical view, refer to the Dynamic Workload Console User's Guide.

Defining event rules based on SAP background events

A scheduling event rule defines a set of actions to run when specific event conditions occur. The definition of an event rule correlates events and triggers actions.

An event rule is identified by a rule name and by a set of attributes that specify if the rule is active, the time frame of its validity, and other information required to decide when actions are triggered. It includes information related to the specific events (`eventCondition`) that the rule must detect and the specific actions it is to trigger upon their detection or timeout (`ruleAction`). Complex rules might include multiple events and multiple actions.

If you are using XBP 3.0, only the SAP background events that are stored in the event history table are considered by IBM Workload Scheduler.

To define event rules, you can use either of the following:

The composer command line

You edit the rules with an XML editor of your choice. For details about how to use the composer to define event rules, see the *IBM Workload Scheduler User's Guide and Reference*.

The Dynamic Workload Console

For information about creating an event rule, see the section about creating an event rule in *Dynamic Workload Console User's Guide*.

For more details about the properties used to define the SAP event rule, see the table available only in html format at the following link: SAP Monitor .

The SAP background event is identified by the following information:

SAP Event ID

The name identifying the SAP event. Wildcards are not allowed.

If you are using the Dynamic Workload Console, you can type the event name in the SAP Event ID field. This field does not support wildcard characters (* and %), nor the following special characters: asterisk (*), question mark (?), and backslash (\). Note that for supported special characters, the escape character (\) must not be used.

Alternatively, you can use the lookup function to search for and select the event name. When specifying the string to search for that represents the SAP Event ID, wildcard characters are supported, (* and %). For example, if you specify "myevent*", then results can include events such as "myevent", "myevent%", and "myevents".

Event parameter

The parameter associated with the SAP event, if any. Wildcards are not allowed.

If you are using the Dynamic Workload Console, the following special characters are not supported when specifying the event parameter: asterisk (*), question mark (?), and backslash (\).

Extended or dynamic agent workstation

The name of the extended agent workstation or the name of the dynamic agent workstation running event monitoring.

Note:

1. If you specify a pattern with the wildcard asterisk (*), all the agents whose name matches the pattern will monitor the specified event.
2. As a best practice, define that an event belonging to an SAP system is monitored by one agent workstation only. If the same SAP event is monitored by more than one agent, you might either be notified multiple times for the same event occurrence or the first agent that notifies the event occurrence makes that event unavailable to the other agents.
3. If you modify the extended agent configuration in the `r3batch` option files, to make the changes effective you must stop and restart the agent.
4. For dynamic agents you can specify the name of a local options file. In the Properties section of the Create Event Rules window of the Dynamic Workload Console a lookup button provides a list of all the local options files associated with that agent. If you do not specify the name of a local options file, the global options file is used by default in the rule definition.

SAP events matching criteria

The SAP background events specified in the event rule are matched with the events raised in the SAP system, according to the following criteria. Depending on the parameters you set:

The SAP event ID and parameter are specified in the event rule

To match, the SAP event ID and parameter must be the same as the event ID and event parameter raised in the SAP system. Also, the event state must be N (New). SAP events with a different parameter or without any parameter are ignored.

The information collected about the matching SAP event is sent by the `r3evmon` process to IBM Workload Scheduler. If the notification is successfully sent, the event is committed on the SAP system and its state changed to C (Confirmed).

For example, you define an event rule in your IBM Workload Scheduler plan based on the following SAP event:

SAP event ID

SAP_TEST

SAP event parameter

ABCDEF

Workstation

An extended agent named GENIUS

According to these settings, a file named `GENIUS_r3evmon.cfg` is created on `GENIUS`. It contains the following `!R3EVENT` keyword:

```
!R3EVENT 0008SAP_TEST0006ABCDEF
```

Monitoring of the `SAP_TEST` event with parameter `ABCDEF` is automatically started. Suppose that the following SAP events were raised on the SAP system:

Table 27. History table of the SAP events raised

EVENT GUID	SAP EVENT ID	EVENT PARM	EVENT SERVER	EVENT TIMESTAMP	EVENT STATE	PROCESS STATE	COUNT OF JOBS
1234	SAP_TEST	ABC123	...	20070925 13:00	C	OK	1
2345	SAP_TEST	ABCD	...	20070925 14:00	N	OK	2
3456	SAP_TEST		...	20070925 15:00	N	OK	3
4567	SAP_TEST	ABCDEF	...	20070925 16:00	N	OK	4

Only the following SAP event is notified to IBM Workload Scheduler:

Table 28. SAP event matching with the event rule defined

EVENT GUID	SAP EVENT ID	EVENT PARM	EVENT SERVER	EVENT TIMESTAMP	EVENT STATE	PROCESS STATE	COUNT OF JOBS
4567	SAP_TEST	ABCDEF	...	20070925 16:00	N	OK	4

If the notification is successfully sent, the event is committed on the SAP system and its state changed to C (Confirmed).

Only the SAP event ID is specified in the event rule

To match, the SAP event ID must be the same as the ID of the events raised in the SAP system whose state is N (New). The parameters of the SAP events, whether specified or not, are not taken into account.

The information collected about all the matching SAP events is sent by the `r3evmon` process to IBM Workload Scheduler. Each event successfully notified is committed on the SAP system and its status changed to C (Confirmed).

For example, you define an event rule in your IBM Workload Scheduler plan based on the following SAP event:

SAP event ID

SAP_TEST

Workstation

GENIUS

According to these settings, a file named `GENIUS_r3evmon.cfg` is created on `GENIUS`. It contains the following `!R3EVENT` keyword:

```
!R3EVENT 0008SAP_TEST
```

Monitoring of the `SAP_TEST` event is automatically started. Suppose that the following SAP events were raised on the SAP system:

Table 29. History table of the SAP events raised

EVENT GUID	SAP EVENT ID	EVENT PARM	EVENT SERVER	EVENT TIMESTAMP	EVENT STATE	PROCESS STATE	COUNT OF JOBS
1234	SAP_TEST	ABC123	...	20070925 13:00	C	OK	1
2345	SAP_TEST	ABCD	...	20070925 14:00	N	OK	2
3456	SAP_TEST		...	20070925 15:00	N	OK	3
4567	SAP_TEST	ABCDEF	...	20070925 16:00	N	OK	4

Only the following SAP events are notified to IBM Workload Scheduler:

Table 30. SAP events matching with the event rule defined

EVENT GUID	SAP EVENT ID	EVENT PARM	EVENT SERVER	EVENT TIMESTAMP	EVENT STATE	PROCESS STATE	COUNT OF JOBS
2345	SAP_TEST	ABCD	...	20070925 14:00	N	OK	2
3456	SAP_TEST		...	20070925 15:00	N	OK	3
4567	SAP_TEST	ABCDEF	...	20070925 16:00	N	OK	4

Each event whose notification is successfully sent is committed on the SAP system and its state changed to C (Confirmed).

Setting a filter for SAP background events in the security file

In the security file, you can filter the SAP background events that can be used to define event rules. By doing this, you restrict the use of certain SAP events to specific users. For example, assume that you want your USA department to manage only the SAP events whose ID begins with `SAP_USA`, and your Italy department to manage all events except those beginning with `SAP_USA`. In the security file that defines the user access for the USA department, define the `CUSTOM` keyword for the `EVENT` object as follows:

```
EVENT      PROVIDER=@ +CUSTOM=SAP_USA@  ACCESS=USE
```

where:

`PROVIDER=@`

Specifies that the user can use the events coming from any provider.

`+CUSTOM=SAP_USA@`

Specifies that the user can use only the SAP events whose ID begins with `SAP_USA`.

This keyword applies only to the SAP provider (SapMonitor).

`ACCESS=USE`

Sets the user access to the object to `USE`.

In the security file that defines the user access for the Italy department, define the `CUSTOM` keyword for the `EVENT` object as follows:

```
EVENT      PROVIDER=@ ~CUSTOM=SAP_USA@  ACCESS=USE
```

where:

`PROVIDER=@`

Specifies that the user can use the events coming from any provider.

`~CUSTOM=SAP_USA@`

Specifies that the user can use all SAP events, except those whose ID begins with `SAP_USA`.

This keyword applies only to the SAP provider (SapMonitor).

`ACCESS=USE`

Sets the user access to the object to `USE`.

For more details about the security file and how to set up user authorizations, see the *IBM Workload Scheduler: Administration Guide*.

Defining event rules based on IDoc records

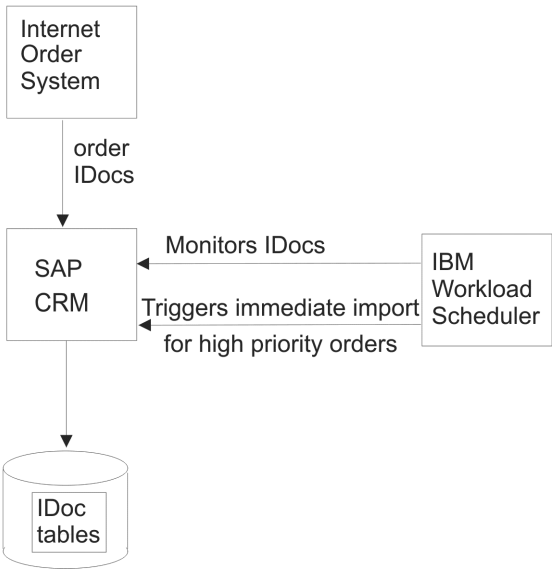
You can use IBM Workload Scheduler to monitor Intermediate Document (IDoc) records in SAP systems and forward events to the IBM Workload Scheduler event integration framework.

To do this, you define an event condition that contains the criteria that the IDocs must match to be forwarded to IBM Workload Scheduler. When the event condition occurs, the action that you associated with it (for example, running a job) is performed.

Business scenario

You connected your Internet sales application to your SAP Customer Relationship Management (CRM) system, which receives the orders as incoming IDocs. The orders are classified as emergency and ordinary, and therefore have different IDoc message types. You want the emergency orders to be imported into the CRM system directly, and the ordinary orders to be processed in batch mode. To do this, in IBM Workload Scheduler, you define an event rule that monitors the IDoc message types corresponding to emergency orders and sends an event to IBM Workload Scheduler. In IBM Workload Scheduler, you define a job to be released when this type of event is received and is linked to an SAP job that runs an import ABAP report for these specific types of IDocs.

Figure 16. Managing high priority IDocs overview



Creating event rules based on IDocs

About this task

To define event rules based on IDocs, specify the fields to be used as matching criteria during IDoc monitoring. For details about these fields, refer to Events matching criteria on page 198. To create the event rules, you can use either of the following:

The composer command line

You edit the rules with an XML editor of your choice. For a general explanation about how to use the composer to define event rules, see the *IBM Workload Scheduler: User's Guide and Reference*. The event condition requires:

- `SAPMonitor` as event monitor provider.
- `IDOCEventGenerated` as event type.

For a list of the values that you can specify in the **attributeFilter** name when defining the event condition, refer to Table 33: Parameters of IDOCEventGenerated event type on page 200.

The Dynamic Workload Console

For information about creating an event rule, see the section about creating an event rule in *Dynamic Workload Console User's Guide*.

For more details about the properties used to define the IDoc event rule, see the following table available only in html format in the online information center: SAP Monitor and browse to the **IDoc Event Raised on XA Workstations** section.

Note:

1. To be able to define and monitor event rules, ensure that you configured your environment as described in Configuring SAP event monitoring on page 100.
2. To configure how IBM Workload Scheduler retrieves the IDoc monitors, set **idoc_no_history** and **idoc_shallow_result** in the options file. For details about these options, refer to Defining the common options on page 85.

Events matching criteria

Table 31: IBM Workload Scheduler fields used to define event rules based on IDocs on page 198 lists the IBM Workload Scheduler fields corresponding to the fields in the IDoc record that you want to search. During monitoring, each IDoc matching the search criteria generates an event that is sent to IBM Workload Scheduler.

Table 31. IBM Workload Scheduler fields used to define event rules based on IDocs

Composer property	Console property	IDoc field
SAPClient	SAP client	MANDT
SAPIDocStatus	Status	STATUS
SAPDirectionIDocTransmission	Direction	DIRECT
SAPReceiverPort	Receiver port	RCVPOR
SAPReceiverPartnerFunction	Receiver partner function	RCVPFC
SAPReceiverPartnerType	Receiver partner type	RCVPRT
SAPReceiverPartnerNumber	Receiver partner number	RCVPRN
SAPSenderPort	Sender port	SNDPOR
SAPSenderPartnerType	Sender partner type	SNDPRT
SAPSenderPartnerFunction	Sender partner function	SNDPFC
SAPSenderPartnerNumber	Sender partner number	SNDPRN
SAPLogicalMessageType	Logical message type	MESTYP
SAPNameOfBasicType	Name of basic type	IDOCTP
SAPLogicalMessageCode	Logical message code	MESCOD
SAPLogicalMessageFunction	Logical message function	MESFCT
SAPTestFlag	Test flag	TEST
SAPOutputMode	Output mode	OUTMOD

Optionally, you can define also correlation rules by using the fields listed in Table 32: IBM Workload Scheduler fields used to define correlation rules for IDoc events on page 199. Date and time values are specified in GMT time zone.

Table 32. IBM Workload Scheduler fields used to define correlation rules for IDoc events

Composer property	Console property	IDoc field
SAPIDocNumber	IDoc number	DOCNUM
SAPReleaseForIDoc	IDoc SAP release	DOCREL
SAPIDocType	IDoc type	DOCTYP
SAPReceiverAddress	Receiver SADR address	RCVSAD
SAPReceiverSADRClient	Receiver SADR client	RCVSMN
SAPFlagForInternationalReceiverAddress	Receiver SADR flag	RCVSNA
SAPReceiverCommunicationType	Receiver SADR communication type	RCVSCA
SAPDefaultFlagForReceiverAddress	Receiver SADR default flag	RCVSDF
SAPReceiverAddressSequentialNumber	Receiver SADR sequential number	RCVSLF
SAPReceiverLogicalAddress	Receiver logical address	RCVLAD
SAPEDIStandard	EDI Standard	STD
SAPEDIStandardVersion	EDI standard version	STDVRS
SAPEDIMessageType	EDI message type	STDMES
SAPSenderAddress	Sender SADR address	SNDSAD
SAPSenderSADRClient	Sender SADR client	SNDSMN
SAPFlagForInternationalSenderAddress	Sender SADR flag	SNDSNA
SAPSenderCommunicationType	Sender SADR communication type	SNDSCA
SAPDefaultFlagForSenderAddress	Sender SADR default flag	SNDSDF
SAPSenderAddressSequentialNumber	Sender SADR sequential number	SNDSLF
SAPSenderLogicalAddress	Sender logical address	SNDLAD
SAPReferenceToInterchangeFile	Interchange file reference	REFINT
SAPReferenceToMessageGroup	Message group reference	REFGRP
SAPReferenceToMessage	Message reference	REFMES
SAPEDIArchiveKey	EDI archive key	ARCKEY
SAPIDocCreationDate	IDoc creation date	CREDAT
SAPIDocCreationTime	IDoc creation time	CRETIM
SAPExtension	Extension	CIMTYP

Table 32. IBM Workload Scheduler fields used to define correlation rules for IDoc events (continued)

Composer property	Console property	IDoc field
SAPEDIALESerializationField	EDI/ALE Serialization field	SERIAL
SAPOverridingInInboundProcessing	Overriding in inbound processing	EXPRSS
SAPIDocChangeDate	IDoc last update date	UPDDAT
SAPIDocChangeTime	IDoc last update time	UPDTIM

Based on the defined rule, the `r3evmon` process of IBM Workload Scheduler monitors the events related to IDoc records according to a polling rate. To customize this polling rate, use the **evmon_interval** option; for details, see Defining the common options on page 85.

Table 33: Parameters of IDOCEventGenerated event type on page 200 lists the values that you can specify as attribute filter name when defining the event condition.

Table 33. Parameters of IDOCEventGenerated event type

Property name	Description	Type	Filtering allowed	Required	Multiple values allowed	Wildcard allowed	Length (min-max)	
SAPClient	SAP client number	numeric (0-9)	✓	✓		✓	1	3
SAPIDocStatus	IDoc status information For a list of allowed values, refer to Table 34: Standard outbound IDoc statuses on page 202 and Table 35: Standard inbound IDoc statuses on page 203.	numeric	✓	✓	✓		1	2
SAPDirectionIDocTransmission	IDoc direction	numeric Value can be 1 (outbound) or 2 (inbound).	✓	✓			1	1
SAPReceiverPort	Receiver port. SAP system, EDI subsystem	string	✓				1	10
SAPReceiverPartnerFunction	Partner function of receiver	string	✓				1	2
SAPReceiverPartnerType	Partner type of receiver	string	✓				1	2
SAPReceiverPartnerNumber	Partner number of receiver	string	✓				1	10

Table 33. Parameters of IDOCEventGenerated event type

(continued)

Property name	Description	Type	Filtering allowed	Required	Multiple values allowed	Wildcard allowed	Length (min-max)	
SAPSenderPort	Sender port. SAP system, EDI subsystem	string	✓				1	10
SAPSenderPartnerType	Partner type of sender	string	✓				1	2
SAPSenderPartnerFunction	Partner function of sender	string	✓				1	2
SAPSenderPartnerNumber	Partner number of sender	string	✓				1	10
SAPLogicalMessageType	Logical message type	string	✓			✓	1	30
SAPNameOfBasicType	Name of basic type	string	✓			✓	1	30
SAPLogicalMessageCode	Logical message code	string	✓				1	3
SAPLogicalMessageFunction	Logical message function	string	✓				1	3
SAPTestFlag	Test flag	string	✓				1	1
SAPOutputMode	Output Mode	string Value can be 2 (immediate sending) or 4 (collected sending).	✓				1	1

Table 34: Standard outbound IDoc statuses on page 202 lists the standard outbound IDoc statuses and Table 35: Standard inbound IDoc statuses on page 203 lists the standard inbound IDoc statuses. Optionally, you can activate a check to prevent event rule definitions with inconsistent IDoc status list and direction. If you activate the check and specify inconsistent values when defining a rule (for example, 02 as status and 2 as direction), you receive an error message and you cannot save the rule definition. To activate the check, perform the following steps:

1. In the *TWA_home*`>\eventPlugIn` directory on Windows™ and in the *TWA_DATA_DIR*`/eventPlugIndirectory` on UNIX®, create the `SapMonitorPlugIn.properties` file.
2. Edit `SapMonitorPlugIn.properties` to set the following configuration property:

   ```
   TWSPlugIn.event.idoc.consistency.check = true
   ```

3. From conman, stop and restart the event processing server by using, respectively, the **stopeventprocessor** and **starteventprocessor** commands.

The default value is `false`.

To have predictable event action results, when defining event rules consider using only non-transitory statuses that allow user checks.

Table 34. Standard outbound IDoc statuses

Status	Description
01	IDoc generated
02	Error passing data to port
03	Data passed to port
04	Error within control information of EDI subsystem
05	Error during translation
06	Translation
07	Error during syntax check
08	Syntax check
09	Error during interchange
10	Interchange handling
11	Error during dispatch
12	Dispatch OK
13	Retransmission OK
14	Interchange acknowledgement positive
15	Interchange acknowledgement negative
16	Functional acknowledgement positive
17	Functional acknowledgement negative
18	Triggering EDI subsystem OK
19	Data transfer for test OK
20	Error triggering EDI subsystem
22	Dispatch OK, acknowledgement still due
23	Error during retransmission
24	Control information of EDI subsystem OK
25	Processing despite syntax error
26	Error during syntax check of IDoc
27	Error in dispatch level (ALE service)
29	Error in ALE service
30	IDoc ready for dispatch (ALE service)
31	Error no further processing

Table 34. Standard outbound IDoc statuses (continued)

Status	Description
32	IDoc was edited
33	Original of an IDoc which was edited
34	Error in control record of IDoc
36	Electronic signature not performed (timeout)
37	IDoc added incorrectly
38	IDoc archived
39	IDoc is in the target system (ALE service)
40	Application document not created in target system
41	Application document created in target system
42	IDoc was created by test transaction

Table 35. Standard inbound IDoc statuses

Status	Description
50	IDoc added
51	Application document not posted
52	Application document not fully posted
53	Application document posted
54	Error during formal application check
55	Formal application check OK
56	IDoc with errors added
57	Error during application check
58	IDoc copy from R/2 connection
60	Error during syntax check of IDoc
61	Processing despite syntax error
62	IDoc passed to application
63	Error passing IDoc to application
64	IDoc ready to be transferred to application
65	Error in ALE service
66	IDoc is waiting for predecessor IDoc (serialization)

Table 35. Standard inbound IDoc statuses (continued)

Status	Description
68	Error - no further processing
69	IDoc was edited
70	Original of an IDoc which was edited
71	IDoc reloaded from archive
73	IDoc archived
74	IDoc was created by test transaction

For example, you define a rule with the following attributes:

Workstation

A dynamic agent named SAPCPU

SAP client number

001

IDoc status list

56,60

IDoc direction

2 (inbound)

After saving the rule according to these settings, when the rule becomes active a file named `SAPCPU_r3evmon.cfg` is created on `SAPCPU`. It contains the following `!IDOC` keyword:

```
!IDOC 0003001000556,600001200000000000000000000000000000000000000000000000000000000000000000
```

IDoc monitoring is automatically started. When the event condition is verified, the action defined in the rule is triggered

For an explanation of the `!IDOC` keyword format, refer to Table 42: Miscellaneous troubleshooting items on page 221.

Examples of event rules based on IDocs

The following example applies to the scenario described in Business scenario on page 196. It shows an event rule that triggers an import ABAP report when an IDoc is added with a message type corresponding to emergency orders.

```
<?xml version="1.0" encoding="UTF-8"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xmlns="http://www.abc.com/xmlns/prod/tws/1.0/event-management/rules"
              xsi:schemaLocation="http://www.abc.com/xmlns/prod/tws/1.0/
                  event-management/rules EventRules.xsd">
        <eventRule name="scenario1_IDoc" ruleType="filter" isDraft="no">
          <eventCondition name="IDocEventRaised1" eventProvider="SapMonitor"
                          eventType="IDocEventGenerated">
             <scope>
                   001 ON SAPCU WITH 2
             <scope>
```

```
             <filteringPredicate> <attributeFilter name="Workstation" operator="eq">
                    <value>SAPCPU</value>
                </attributeFilter>
                <attributeFilter name="SAPClient" operator="eq">
                    <value>001</value>
                </attributeFilter>
                <attributeFilter name="SAPIDocStatus" operator="eq">
                    <value>50</value>
                </attributeFilter>
                <attributeFilter name="SAPDirectionIDocTransmission" operator="eq">
                    <value>2</value>
                </attributeFilter>
                <attributeFilter name="SAPLogicalMessageType" operator="eq">
                    <value>EORD1</value>
                </attributeFilter>
            </filteringPredicate>
          </eventCondition>
          <action actionProvider="TWSaction" actionType="sbj"
                  responseType="onDetection">
            <description>Trigger immediate report for high priority orders
            </description>
            <parameter name="JobDefinitionWorkstationName">
              <value>MASTER84</value>
            </parameter>
            <parameter name="JobDefinitionName">
              <value>triggerimport</value>
            </parameter>
          </action>
        </eventRule>
</eventRuleSet>
```

The following example shows an event rule defined to create a ticket for failing IDocs in the SAP Solution Manager or any other problem management system: when an IDoc with a syntax error is detected, the engine submits a job to create a ticket for the failing IDoc.

```
<?xml version="1.0" encoding="UTF-8"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xmlns="http://www.abc.com/xmlns/prod/tws/1.0/event-management/rules"
              xsi:schemaLocation="http://www.abc.com/xmlns/prod/tws/1.0/
                  event-management/rules EventRules.xsd">
        <eventRule name="scenario1_IDoc" ruleType="filter" isDraft="no">
          <eventCondition name="IDocEventRaised1" eventProvider="SapMonitor"
                          eventType="IDocEventGenerated">
             <filteringPredicate> attributeFilter name="Workstation" operator="eq">
                    <value>SAPCPU</value>
                </attributeFilter>
                <attributeFilter name="SAPClient" operator="eq">
                    <value>001</value>
                </attributeFilter>
                <attributeFilter name="SAPIDocStatus" operator="eq">
                    <value>60</value>
                <attributeFilter name="SAPDirectionIDocTransmission" operator="eq">
                    <value>2</value>
                </attributeFilter>
                </attributeFilter>
                <attributeFilter name="SAPLogicalMessageType" operator="eq">
                    <value>MYORD1</value>
                </attributeFilter>
```

```
            </filteringPredicate>
          </eventCondition>
          <action actionProvider="TWSaction" actionType="sbj"
                  responseType="onDetection">
            <description>Create a ticket for failing IDocs
            </description>
            <parameter name="JobDefinitionWorkstationName">
              <value>MASTER84</value>
            </parameter>
            <parameter name="JobDefinitionName">
              <value>createticket</value>
            </parameter>
          </action>
        </eventRule></eventRuleSet>
```

Defining event rules based on CCMS Monitoring Architecture alerts

Use CCMS functions to check the performance of the various SAP system components, diagnose potential problems, and be alerted about error and warning conditions.

The SAP Computing Center Monitoring System (CCMS) is a centralized monitoring architecture that provides a set of monitors for monitoring the SAP environment. Using the CCMS functions you can check the performance of the various SAP system components, diagnose potential problems, and be alerted about error and warning conditions. The monitors provide you with the information you require to fine tune the SAP system and the operating modes, and hence optimize system performance.

With IBM Workload Scheduler, you can integrate the CCMS monitoring functions into your management infrastructure by defining event rules based on the alerts raised in the SAP system.

Business scenarios

The following sections describe:

- Business scenario: defining an event rule to process alerts related to IDocs on page 206
- Business scenario: defining an event rule to process alerts related to operating system on page 207

Business scenario: defining an event rule to process alerts related to IDocs

You connected your Internet sales application to your SAP Customer Relationship Management (CRM) system, which receives the orders as incoming IDocs. You want to import the orders into the CRM system when their number exceeds a specified threshold, therefore you configured your SAP CCMS monitoring architecture to generate an alert when the number of incoming IDocs exceeds a certain value. To automatically start a task that imports the orders:

1. In your SAP CCMS monitoring architecture, identify the element related to the alert that you configured for the incoming order IDocs.
2. In IBM Workload Scheduler, define an event rule, to be active during the timeframe when inbound order traffic is heavy, which monitors the element identified in step 1. As soon as an alert is generated for the element, a CCMS event is sent to IBM Workload Scheduler.
3. In IBM Workload Scheduler, define a job to be submitted when the CCMS event is received, to run an SAP job that runs an import ABAP report for the order IDocs.

Business scenario: defining an event rule to process alerts related to operating system

As an IBM Workload Scheduler administrator, you are in charge of taking the appropriate action in the IBM Workload Scheduler plan when a critical situation occurs in the SAP system. You have an SAP extended agent workstation dedicated to submit Business Intelligence tasks, whose activity you want to suspend every time the SAP BI system faces a critical situation (for example, the SAP system is running out of space). To do this:

1. In your SAP CCMS monitoring architecture, identify the element related to the SAP system you want to monitor.
2. In IBM Workload Scheduler, define an event rule that monitors the element and sends an event to IBM Workload Scheduler when an alert is generated for it. Associate with this event an action that sets the limit of the agent workstation to 0, and sends a mail to the SAP administrator to notify the details of the critical situation.
3. As soon as the SAP administrator resolves the problem, you set the limit of the agent workstation back to its original value to resume the scheduling activities.

Creating event rules based on CCMS alerts

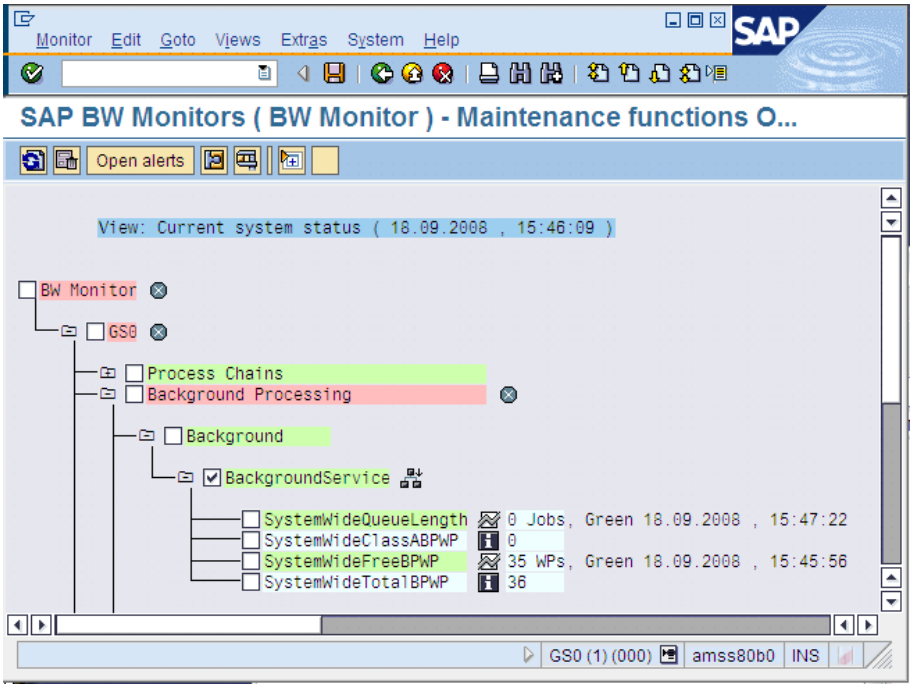
About this task

SAP systems are shipped with a predefined set of monitors, grouped in *monitor sets*. A monitor set contains a list of monitors, each monitor contains a set of *monitoring trees*. A monitor is a set of *monitoring tree elements* (MTEs) that are arranged in a hierarchical structure, named *alert monitoring tree*. You can define event rules based on the alert generated for a specific MTE.

Note: To be able to define and monitor event rules, ensure that you configured your environment as described in Configuring SAP event monitoring on page 100.

Figure 17: A monitor and its MTEs - © SAP AG 2009. All rights reserved. on page 208 shows the monitor named **BW Monitor** (belonging to the monitor set **SAP BW Monitor**) and its associated monitor tree elements (MTEs).

Figure 17. A monitor and its MTEs - © SAP AG 2009. All rights reserved.



To configure how IBM Workload Scheduler retrieves the CCMS alerts, set **ccms_alert_history** in the options file. For details about this option, refer to Defining the common options on page 85.

To create event rules, you can use either of the following:

The composer command line

You edit the rules with an XML editor of your choice. For a general explanation about how to use the composer to define event rules, see the *User's Guide and Reference*.

The Dynamic Workload Console

For information about creating an event rule, see the section about creating an event rule in *Dynamic Workload Console User's Guide*.

For more details about the properties used to define the CCMS event rule, see the following table available only in html format in the online information center: SAP Monitor and browse to the **CCMS Event Raised on XA Workstations** section.

To define the CCMS event for your rule, specify the following information. For more details about how you separate the MTE name into the individual IBM Workload Scheduler fields, see Mapping between the MTE name and IBM Workload Scheduler fields on page 210.

Extended or dynamic agent workstation

The name of the extended agent workstation or the name of the dynamic agent workstation running event monitoring.

Note:

1. If you specify a pattern with the wildcard asterisk (*), all the agents whose name matches the pattern will monitor the specified event.
2. As a best practice, define that an event belonging to an SAP system is monitored by one agent workstation only. If the same SAP event is monitored by more than one agent, you might either be notified multiple times for the same event occurrence or the first agent that notifies the event occurrence makes that event unavailable to the other agents.
3. If you modify the extended agent configuration in the `r3batch` option files, to make the changes effective you must stop and restart the agent.
4. For dynamic agents you can specify the name of a local options file. In the Properties section of the Create Event Rules window of the Dynamic Workload Console a lookup button provides a list of all the local options files associated with that agent. If you do not specify the name of a local options file, the global options file is used by default in the rule definition.

MTE SAP System ID

Name of the SAP system where the MTE is located (for example, `GS0` in Figure 17: A monitor and its MTEs - © SAP AG 2009. All rights reserved. on page 208). This field is required. Wildcards are not allowed, you can specify up to eight characters.

MTE Monitoring Context Name

Name of the monitoring context to which the MTE belongs. This field is required. A monitoring context is a logically connected group of monitoring objects that are ordered together under one summary in the monitoring tree (for example, `Background` in Figure 17: A monitor and its MTEs - © SAP AG 2009. All rights reserved. on page 208).

Wildcards are not allowed, you can specify up to 40 characters.

MTE Monitoring Object Name

Name of the monitoring object in the alert monitor. This field is required. A monitoring object is a component or property of the system that is to be monitored (for example, `BackgroundService` in Figure 17: A monitor and its MTEs - © SAP AG 2009. All rights reserved. on page 208). If you choose not to specify a value, you must leave the value NULL, which is the default.

Wildcards are not allowed, you can specify up to 40 characters.

MTE Monitoring Attribute Name

Name of the monitoring attribute in the alert monitor. In the monitoring tree, a monitoring attribute is always an end node in the hierarchy (for example, `SystemWideFreeBPWP` in Figure 17: A monitor and its MTEs - © SAP AG 2009. All rights reserved. on page 208). This field is required. If you choose not to specify a value, you must leave the value NULL, which is the default.

Wildcards are not allowed, you can specify up to 40 characters.

Alert Value

Numeric value that indicates the color of the alert generated for the MTE. This field is optional. You can specify one or a combination of the following values:

1

Green, meaning Everything OK.

2

Yellow, meaning Warning.

3

Red, meaning Problem or error.

If you do not specify any value, all the alerts generated for the MTE are considered.

Alert Severity

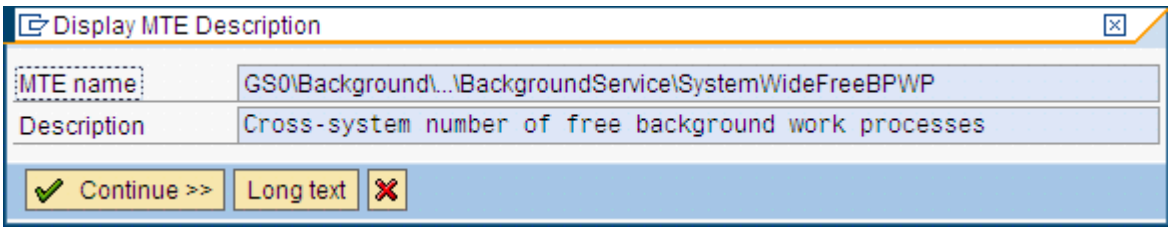
Severity of the alert. It can be a number between 0 (lowest) and 255 (highest), or a range among these values. This field is optional. Alert severity is assigned during alert configuration; the SAP standard configuration is 50.

Mapping between the MTE name and IBM Workload Scheduler fields

About this task

Within SAP, MTEs are identified by a name made up of several tokens, separated by backslashes (\). To display the complete MTE name, select the MTE and click **Properties** or press **F1**:

Figure 18. Name and description of an MTE - © SAP AG 2009. All rights reserved.



According to the type of MTE that you want to monitor, you must fill in each IBM Workload Scheduler field with a specific token of the MTE name (to know your MTE type, select the MTE and click **Legend**):

If you are using Dynamic Workload Console V8.5.1, or later

1. In the **Event Rule Editor** panel, from the **Properties** section, click **Autofill MTE Tokens**. The **MTE Name** window opens.
2. In the **MTE Name** field, write the name of the MTE to monitor and click **OK**. You are returned to the **Event Rule Editor** panel, where the IBM Workload Scheduler fields are filled in accordingly.

If you are using Dynamic Workload Console prior to V8.5.1

Refer to the instructions provided in the following sections:

- Context MTE on page 211
- Object MTE on page 212
- Attribute MTE on page 212

Note: Virtual MTEs cannot be monitored.

Context MTE

A context MTE is the uppermost node of a monitoring tree; it contains all the associated object MTEs and attribute MTEs. Context nodes can be either of the following types:

Root

Belongs only to the All Monitoring Contexts monitor. According to the SAP version you are using, a root context MTE name can have either of the following formats:

```
tokenA\tokenB\...
- OR -
tokenA\tokenB
```

For example:

```
T10\SystemConfiguration\...
```

Refer to Table 36: Mapping between root context MTE name and IBM Workload Scheduler fields on page 211 for an explanation about how you report this type of MTE in the IBM Workload Scheduler fields:

Table 36. Mapping between root context MTE name and IBM Workload Scheduler fields

IBM Workload Scheduler field	Token of MTE name	In this example...
MTE SAP System ID	*tokenA*	`T10`
MTE Monitoring Context Name	*tokenB*	`SystemConfiguration`
MTE Monitoring Object Name	N/A	`NULL`
MTE Monitoring Attribute Name	N/A	`NULL`

Summary

According to the SAP version you are using, a summary context MTE name can have either of the following formats:

```
tokenA\tokenB\...\tokenC\...
- OR -
tokenA\tokenB\tokenC
```

For example:

```
T10\SystemConfiguration\...\InstalledSupportPackages\...
```

Refer to Table 37: Mapping between summary context MTE name and IBM Workload Scheduler fields on page 212 for an explanation about how you report this type of MTE in the IBM Workload Scheduler fields:

Table 37. Mapping between summary context MTE name and IBM Workload Scheduler fields

IBM Workload Scheduler field	Token of MTE name	In this example...
MTE SAP System ID	*tokenA*	`T10`
MTE Monitoring Context Name	*tokenB*	`SystemConfiguration`
MTE Monitoring Object Name	*tokenC*	`InstalledSupportPackages`
MTE Monitoring Attribute Name	N/A	`NULL`

Object MTE

According to the SAP version you are using, an object MTE name can have either of the following formats:

```
tokenA\tokenB\tokenC\tokenD
- OR -
tokenA\tokenB\...\tokenD
```

For example:

```
PR0\amsp53_PR0_11\R3Services\Background\
```

Refer to Table 38: Mapping between object MTE name and IBM Workload Scheduler fields on page 212 for an explanation about how you report this type of MTE in the IBM Workload Scheduler fields:

Table 38. Mapping between object MTE name and IBM Workload Scheduler fields

IBM Workload Scheduler field	Token of MTE name	In this example...
MTE SAP System ID	*tokenA*	`PR0`
MTE Monitoring Context Name	*tokenB*	`amsp53_PR0_11`
MTE Monitoring Object Name	*tokenD*	`Background`
MTE Monitoring Attribute Name	N/A	`NULL`

Attribute MTE

According to the SAP version you are using, an attribute MTE name can have either of the following formats:

```
tokenA\tokenB\tokenC\tokenD\tokenE
- OR -
tokenA\tokenB\...\tokenD\tokenE
```

For example:

```
PR0\amsp53_PR0_11\R3Services\Background\AbortedJobs
```

Refer to Table 39: Mapping between attribute MTE name and IBM Workload Scheduler fields on page 213 for an explanation about how you report this type of MTE in the IBM Workload Scheduler fields:

Table 39. Mapping between attribute MTE name and IBM Workload Scheduler fields

IBM Workload Scheduler field	Token of MTE name	In this example...
MTE SAP System ID	*tokenA*	`PR0`
MTE Monitoring Context Name	*tokenB*	`amsp53_PR0_11`
MTE Monitoring Object Name	*tokenD*	`Background`
MTE Monitoring Attribute Name	*tokenE*	`AbortedJobs`

Setting correlation rules and action parameters

Optionally, you can use the alert properties listed in Table 40: Alert properties for correlations on page 213 to:

- Define correlation rules between CCMS events.
- Specify additional parameters for the action that is associated with the event rule.

Date and time values are specified in GMT time zone.

Table 40. Alert properties for correlations

Table 40. Alert properties for correlations (continued)

CCMS alert property	Console property	Composer property
MSGCLASS	XMI Ext Company Name	XMIExtCompanyName
MSGID	XMI Log Msg ID	XMILogMsgID
MTCLASS	Alert MT Class	AlertMTClass
MTINDEX	Alert MT Index	AlertMTIndex
MTMCNAME	Alert Monitoring Context Name	AlertMTEContext
MTNUMRANGE	Alert MTE Range	AlertMTERange
MTSYSID	Alert MTE System	AlertMTESys
MTUID	Alert MT Type ID	AlertMTTypeID
OBJECTNAME	Alert Monitoring Object Name	AlertMonObjName
RC	Alert Return Code	AlertReturnCode
REPORTEDBY	Alert Reported By	AlertReportedBy
SEVERITY	Alert Severity	AlertSeverity
STATCHGBY	Alert Changed By	AlertChangedBy
STATCHGDAT	Alert Change Date	AlertChangeDate
STATCHGTIM	Alert Change Time	AlertChangeTime
STATUS	Alert Status	AlertStatus
USERID	User ID	UserID
VALUE	Alert Value	AlertValue

Getting alert status and committing alerts by an external task

Learn how to get CCMS alert status and commit CCMS alerts.

Refer to the following sections for details about:

- Getting CCMS alert status on page 214
- Committing CCMS alerts on page 216

Getting CCMS alert status

About this task

To get the current status of a CCMS alert from IBM Workload Scheduler, use the external task Get Information (GI). To replace the command arguments with the actual values, refer to the output returned by the event rule you defined. For details

about the correspondence between the CCMS properties and the Console and composer properties, see Table 40: Alert properties for correlations on page 213.

From a command line, enter the following command:

The following is an example of how to retrieve the current status of a CCMS alert:

```
r3batch -t GI -c horse10 -- " -t GAS -alsysid T10
        -msegname SAP_CCMS_horse10_T10_00 -aluniqnum 0017780869
        -alindex 0000000104 -alertdate 20081007 -alerttime 040356"
```

You are returned the current status of the alert.

Committing CCMS alerts

About this task

The CCMS alerts that you defined as IBM Workload Scheduler events are not automatically committed after their processing. To commit an alert that was processed by IBM Workload Scheduler, use the external task Put Information (PI).

To replace the command arguments with the actual values, refer to the output returned by the event rule you defined. For details about the correspondence between the CCMS properties and the Console and composer properties, see Table 40: Alert properties for correlations on page 213.

From a command line, enter the following command:

Command syntax

```
-r3batch -t PI -c XA_Unique_ID -- " -t CA -alsysidsap_system_ID -msegnamealert_mte_segment -aluniqnumalert_UID -alindexalert_index -alertdatealert_date -alerttimealert_time "
```

Where:

-t PI

Identifier of the task to be performed, in this case PI (Put Information).

-c *Agent_name*

The name of the dynamic agent workstation or the unique identifier for the extended agent workstation connected to the SAP system where the MTE for which the alert was raised is located. For information about retrieving the unique identifier for the extended agent workstation, see UNIQUE_ID on page 21.

-t CA

Identifier of the task to be performed, in this case CA (Commit Alert).

-alsysid *sap_system_ID*

Identifier of the SAP system where the MTE for which the alert was raised is located. If the name contains blanks, enclose it between single quotes.

-msegname *alert_monitoring_segment*

Name of the alert monitoring segment. You can specify from 1 to 40 characters.

-aluniqnum *alert_UID*

Unique identifier of the alert, made up of 10 characters.

-alindex *alert_index*

Alert index, made up of 10 characters.

-alertdate *alert_date*

Date of the alert, in the format *yyyymmdd*.

-alerttime *alert_time*

Time of the alert, in the format *hhmmss*.

The following is an example of how to commit a CCMS alert:

```
r3batch -t PI -c horse10 -- " -t CA -alsysid T10
        -msegname SAP_CCMS_horse10_T10_00 -aluniqnum 0017780869
        -alindex 0000000104 -alertdate 20081007 -alerttime 040356"
```

You are returned with the message `The CCMS alert was successfully confirmed`.

Example of an event rule based on CCMS alerts

The following example shows an event rule defined to monitor the yellow alerts raised on the MTE named `GS0\ALE/EDI GS0(000) Log.sys TVALE\Inbound IDoc ORDER_IDOC\Inbound: IDoc generated`. The MTE is configured to generate a yellow alert when the number of IDocs representing orders ready to process exceeds a specified threshold. If this condition occurs, the following actions are triggered:

- An IBM Workload Scheduler job is submitted to process the order IDocs.
- An IBM Workload Scheduler job, with priority 10, is submitted to confirm the alert.

```
<?xml version="1.0" encoding="UTF-8"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.abc.com/xmlns/prod/tws/1.0/event-management/rules"
  xsi:schemaLocation="http://www.abc.com/xmlns/prod/tws/1.0/event-management/rules
      http://www.abc.com/xmlns/prod/tws/1.0/event-management/rules/EventRules.xsd">
     <eventRule name="SCENARIO1_XAL" ruleType="filter" isDraft="yes">
     <eventCondition name="MTEEventRaised1" eventProvider="SapMonitor"
                     eventType="CCMSEventGenerated">
         <filteringPredicate>
              <attributeFilter name="Workstation" operator="eq"
                    <value>SAP_XA</value>
              </attributeFilter>
              <attributeFilter name="InputSAPSystemID" operator="eq"
                    <value>GS0</value>
              </attributeFilter>
              <attributeFilter name="InputMonitoringContextName" operator="eq"
                    <value>ALE/EDI GS0(000) Log.sys TVALE</value>
              </attributeFilter>
              <attributeFilter name="InputMonObjectName" operator="eq"
                    <value>Inbound IDoc ORDER_IDOC</value>
              </attributeFilter>
              <attributeFilter name="InputMonFieldName" operator="eq"
                    <value>Inbound: IDoc generated</value>
              </attributeFilter>
         </filteringPredicate>
     </eventCondition>
     <action actionProvider="TWSAction" actionType="sbj" responseType="onDetection">
              <parameter name="JobUseUniqueAlias">
                    <value>false</value>
              </parameter>
              <parameter name="JobDefinitionWorkstationName">
                    <value>SAP_XA/value>
              </parameter>
```

```
            <parameter name="JobAlias">
                  <value>IDOC_%{MTEEventRaised1.AlertUID}/value>
            </parameter>
            <parameter name="JobDefinitionName">
                  <value>PROCESS_ORDER/value>
            </parameter>
     </action>
     <action actionProvider="TWSAction" actionType="sbd" responseType="onDetection">
            <parameter name="JobUseUniqueAlias">
                  <value>false/value>
            </parameter>
            <parameter name="JobWorkstationName">
                  <value>TWS_HOST_FTA/value>
            </parameter>
            <parameter name="JobTask">
                  <value>C:\TWA_home\methods\r3batch -t PI
                        -c %{MTEEventRaised1.Workstation} -- "
                        -t CA -ALSYSID %{MTEEventRaised1.AlertSAPSystemID}
                        -MSEGNAME %{MTEEventRaised1.AlertMTESegment}
                        -ALUNIQNUM %{MTEEventRaised1.AlertUID}
                        -ALINDEX %{MTEEventRaised1.AlertIndex}
                        -ALERTDATE %{MTEEventRaised1.AlertDate}
                        -ALERTTIME %{MTEEventRaised1.AlertTime} "
                  </value>
             </parameter>
            <parameter name="JobPriority">
                  <value>10</value>
            </parameter>
            <parameter name="JobType">
                  <value>Command</value>
            </parameter>
            <parameter name="JobAlias">
                  <value>CONFIRM_%{MTEEventRaised1.AlertUID}</value>
            </parameter>
            <parameter name="JobStreamName">
                  <value>CONFIRM_STREAM</value>
            </parameter>
            <parameter name="JobLogin">
                  <value>twsuser</value>
            </parameter>
     </action>
     </eventRule>
</eventRuleSet>
```

National Language support

The National Language support feature allows you to install SAP on a localized IBM Workload Scheduler workstation and use localized characters for IBM Workload Scheduler job names, job streams, and SAP variants.

Using the local and global configuration files, you can set up SAP to use different code pages and languages for both its output and its connection with a remote SAP system.

As described in Unicode support on page 75, this version of Access method for SAP features Unicode, which is widely supported by SAP systems since version 4.7. However, if either the workstation running SAP or the target SAP systems do not support Unicode, this section describes how you configure code pages and national languages for SAP.

Setting National Language support options

The following options control the code page and national language used by Access method for SAP, when Unicode support is not used:

TWSXA_CP

The code page used to establish the connection between `r3batch` and the target SAP system.

If you are running a non-Unicode version of `r3batch`, set this option to the code page installed on the SAP system (for a list of the valid code pages, refer to SAP supported code pages on page 220). The default value is the SAP code page 1100, similar to the standard ISO8859-1. In all other cases, this option is ignored.

TWSXA_LANG

The language that `r3batch` uses to log in. It can be one of the following (DE, EN, and JA can be set from the Option Editor. The other languages can be set using any text editor):

- Brazilian Portuguese (pt_BR)
- English (EN, the default value)
- French (FR)
- German (DE)
- Italian (IT)
- Japanese (JA)
- Korean (KO)
- Simplified Chinese (zh_CN)
- Spanish (ES)
- Traditional Chinese (zh_TW)

TWSMETH_CP

The code page that `r3batch` uses for its output. The default is the code page used by the IBM Workload Scheduler workstation that hosts `r3batch`.

Ensure that the `TWSMETH_CP` and `TWSMETH_LANG` options are consistent.

TWSMETH_LANG

The catalog language used by `r3batch`. The default is the language used by the IBM Workload Scheduler workstation that hosts `r3batch`.

Ensure that the `TWSMETH_CP` and `TWSMETH_LANG` options are consistent.

SAP supported code pages

To communicate with SAP systems, Access method for SAP uses the following code pages. Use these values to set option `TWSXA_CP`, only when `r3batch` does not support Unicode.

Table 41. SAP supported code pages

SAP code pages	Description
1100	8859-1, this is the default value
1103	MS 850
8000	SJIS: Shift JIS
8300	BIG5: Traditional Chinese
8400	GBK: Simplified Chinese

Troubleshooting

Learn what to do if you get any problems while installing or using IBM Workload Scheduler access methods or plug-ins.

Troubleshooting the SAP connection

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

- Ensure that you can ping the SAP system from the IBM Workload Scheduler system. This shows basic network connectivity.
- Note that using the SAP routers to access the R/3 system could exceed the size of internal buffers of the RFC library used to store the hostname of the SAP system. When this occurs, the hostname gets truncated, causing the connection to the R/3 system to fail. To work around this problem, do not fully qualify the name of the SAP routers or alternatively use the IP addresses.
- Run the following **telnet** command to verify connectivity:

  ```
  telnet systemname 33xx
  ```

  where *systemname* is the system name or IP address of the SAP server and *xx* is the SAP instance.

  If the command fails to complete, this means that communication between `r3batch` and the SAP application server is down.

- Log on to the SAP system as an administrator and verify that the IBM Workload Scheduler RFC user (created in the Creating the IBM Workload Scheduler RFC user on page 65) exists.
- If the SAP gateway truncates the connection string, replace the host name with the IP address.
- If `r3batch` runs on an AIX® system that does not use U.S. English, make sure that the U.S. Language Environment® is installed on both the IBM Workload Scheduler workstation and the SAP database workstation. Otherwise the error `BAD TEXTENV` (or a similar error message) might appear in the `dev_rfc` trace file and connections to SAP fail.

Other known problems

Table 42: Miscellaneous troubleshooting items on page 221 lists miscellaneous troubleshooting problems.

Table 42. Miscellaneous troubleshooting items

Area	Item
`r3batch` and `r3event`: output contains unreadable characters	**Symptom**: When you enter the `r3batch` and `r3event` commands interactively (for example, to export an SAP calendar) the output is returned in UTF-8 format.
	Solution: To resolve this problem, you can either use a shell that supports the UTF-8 code page or redirect the output to a file and open it with a text editor that supports the UTF-8 format.
`r3batch:` SAP jobs contain quotation marks (") or reverse quotes (`)	**Symptoms**: SAP jobs whose names contain quotation marks or reverse quotes are not displayed in the pick list of the Dynamic Workload Console. -OR- You have an IBM Workload Scheduler job that tries to submit an SAP job whose name contains quotation marks, but it abends with an error. The following message might be displayed: `EEWO0439E The required options are not specified either in the global or in the local options file.`
	Solution: In your SAP system, make a copy of the SAP job and assign it a name that does not contain quotation marks or reverse quotes.
`r3batch:` SAP job containing Arabic characters.	**Symptom**: An SAP job abends when the job contains Arabic characters.
	Solution: If you run an SAP job that contains Arabic characters, you must set the local codepage of the agent workstation hosting the r3batch access method to the Arabic codepage. Refer to the **twsmeth_cp** keyword in the common options file, Defining the common options on page 85.

`r3batch:` error messages submitting a job on dynamic agents.	**Symptom**: When working with dynamic workstations and performing actions such as: displaying a process chain, restarting a process chain, or retrieving the spool list, the following messages might be displayed from the Dynamic Workload Console:
	`EEWO0439E The required options are not specified either in the global or in the local options file.`
	`EEWO1065W The environment variable UNISON_JOB is not set. The process chain cannot be restarted.`

	Solution: These messages might indicate that the requested action is not supported on dynamic workstations. Refer to the *IBM Workload Scheduler Release Notes*® for more information about IBM Workload Scheduler features and minimum required versions for compatibility.

Table 42. Miscellaneous troubleshooting items

(continued)

Area	Item
`r3batch:` r3batch hangs when performing actions from the Dynamic Workload Console.	**Symptoms**: r3batch hangs when performing actions from the Dynamic Workload Console such selecting from a pick list, submitting a job, or similar actions that require connection to the SAP system. The IBM Workload Scheduler joblog might also contain multiple "Timer expired" messages.
	Solution: This problem is caused by the IBM Workload Scheduler logging and tracing component. There are two possible solutions: • Deactivate the tracing utility as described in the following technote: http://www.ibm.com/support/docview.wss?uid=swg21503284. OR • Modify the `r3batch.properties` files. Locate the**r3batch.trace.handlers.traceFile.MPFileSemKeyproperties** setting, and then either comment this property setting out or use a different value. Choose any numeric value and retry the operation.
`r3batch:` Submit same process chain in parallel fails.	**Symptom**: The SAP system returns an error message RFC_ERROR_SYSTEM_FAILURE when starting an SAP process chain.
	Solution: Verify if the corrections stated in SAP note 1723482 are applied to your SAP Business Warehouse system or avoid running the same process chain more than once simultaneously.
r3batch: When you restart the process of a subchain, the status of the original process chain is not changed to active	**Symptom**: When you restart the process of a subchain, the status of the original process chain is not changed to active.
	Solution: Refer to SAP Note 1075876.
`r3batch:` Refresh an SAP process chain after a kill action on a running job instance.	**Symptom**: If you perform a kill action on an IBM Workload Scheduler job instance running on a dynamic workstation which monitors an SAP process chain, and then subsequently perform a Refresh operation on this job, the job fails.
	Solution: You cannot perform a Refresh operation after having performed a kill action on an IBM Workload Scheduler job instance running on a dynamic workstation which monitors an SAP process chain. Verify the status of the SAP process chain on the SAP system, and then set the IBM Workload Scheduler job status accordingly.

Table 42. Miscellaneous troubleshooting items

(continued)

Area	Item
`r3batch:` Wrong characters are displayed in the Criteria Manager profile description.	**Symptom**: Special characters such as, < (less than), > (greater than), or the ' (apostrophe) specified in the **Description** field of the **Create criteria profile** dialog are displayed incorrectly.
	Solution: Avoid using special characters in the **Description** field when creating a new criteria profile.
`r3evmon:` monitoring events is not started, stopped, or performed	**Symptom**: You cannot start or stop event monitoring, or event monitoring is not performed.
	Solution: Ensure that *TWSuser* is the owner of the following files, and that the user has read and write permissions: • `/TWA_DATA_DIR/pids/XAname_r3evmon.pid` • `/TWA_DATA_DIR/EIF/XAname_r3evmoncache.dat` • `/TWA_DATA_DIR/EIF/XAname_r3evmoneif.conf` • `/TWA_DATA_DIR/methods/r3evmon_cfg/XAname_r3evmon.cfg` • `/TWA_DATA_DIR/methods/r3evmon_cfg/XAname_r3idocmon.cfg` • `/TWA_DATA_DIR/methods/r3evmon_cfg/XAname_r3xalmon.cfg` • `/TWA_DATA_DIR/methods/r3evmon_cfg/XAname_r3evmon.lck` On Windows™ workstations, these files are located in the TWA_home directory and not in the *TWA_DATA_DIR* directory.
`r3batch:` monitoring SAP events is not performed	**Symptom**: The SAP event on which the event rule is based is neither monitored nor committed.
	Solution: Ensure that the extended agent workstation you specified in the SAP event definition exists. When you define an SAP event within an event rule, no check on the extended agent workstation is made: if the workstation does not exist, the event rule is saved and activated but it will never be resolved.
`r3batch:` monitoring SAP events is not performed	**Symptom**: With XBP 3.0, the SAP event is raised but IBM Workload Scheduler is not notified and therefore does not act as expected.
	Solution: Ensure that the SAP event was not excluded from logging in the SAP event history table.
`r3batch:` monitoring SAP events is not performed	**Symptom**: The SAP events on which the event rule is based are not monitored nor committed.
	Solution: The SAP events being monitored are listed in the following file: `TWA_DATA_DIR/monconf/XAname_r3evmon.cfg` where `XAname` is the name of the SAP extended agent workstation.

Table 42. Miscellaneous troubleshooting items

(continued)

Area	Item
	Check that the file is updated and contains the current monitoring plan. The SAP events are indicated by the following keyword (one for each SAP event on the same extended agent): `!R3EVENT` *SAP_event_name_lengthSAP_event_name*[*SAP_event_parm_lengthSAP_event_parm*] where: ***SAP_event_name_length*** The length of the SAP event name to monitor, in the format *nnnn*. For example, `0008`, if the event name is `SAP_TEST`. ***SAP_event_name*** The name of the SAP event to monitor. ***SAP_event_parm_length*** The length of the parameter associated with the SAP event to monitor, if any. The format is *nnnn*. For example, `0007`, if the event name is `SAP_PAR`. ***SAP_event_parm*** The parameter associated with the SAP event to monitor, if any. This value is optional, but omitting it identifies an SAP event with no parameter associated. For details about how the events are matched between `r3evmon.cfg` and the SAP system, see SAP events matching criteria on page 193. For each configuration file, an `r3evmon` process is started to monitor the SAP events listed. To start an `r3evmon` monitoring process for a specific extended agent workstation, enter the following command. **Note:** 1. For UNIX® only, `r3evmon` must be entered by the owner of the IBM Workload Scheduler installation: 2. If you run `r3evmon` from a Windows™ DOS shell, the command prompt is not returned until the process completes. `r3evmon -t SEM -c` *XA_Unique_ID* `-- "[-EIFSRV` *EIF_server* `-EIFPORT` *EIF_port*`]"` where:

Table 42. Miscellaneous troubleshooting items

(continued)

Area	Item
	XA_Unique_ID The unique identifier of the extended agent workstation. For information about retrieving the unique identifier for the extended agent workstation, see UNIQUE_ID on page 21. ***EIF_server*** The host name or IP address of the master domain manager. ***EIF_port*** The port that the master domain manager uses to receive the event notification.
`r3batch:`IDoc monitoring is not performed	**Symptom**: The events on which the event rule is based are not monitored or no event is generated during IDoc monitoring.

	Solution: The events being monitored are listed in the following file:
	`TWA_DATA_DIR/monconf/XAname_r3evmon.cfg`
	where `XAname` is the name of the SAP extended agent workstation. It is the same file that is used to monitor SAP events in general.
	Check that the file is updated and contains the current monitoring plan. The events corresponding to the `IDOCEventGenerated` event type are indicated by the following keyword (one for each event on the same extended agent):
	`!IDOC nnnn<Client Number>nnnn<IDoc Status List>nnnn<Direction>nnnn<Receiver Port>`
	`nnnn<Receiver Partner Type>nnnn<Partner Function of Receiver>`
	`nnnn<Partner Number of Receiver>nnnn<Sender Port>nnnn<Sender Partner Type>`
	`nnnn<Partner Function of Sender>nnnn<Partner Number of Sender>`
	`nnnn<Message Type>nnnn<IDoc Type>nnnn<Logical Message Variant>`
	`nnnn<Logical Message Function>nnnn<Test Flag>nnnn<Output Mode>`
	where:
	nnnn
	The length of the IDoc field. For example, `0005` indicates the value of an IDoc status list corresponding to `56,60`.
	< >
	Contains the value of the field associated with the IDoc to be monitored. For a list of the supported IDoc fields, refer to Table 31: IBM Workload Scheduler fields used to define event rules based on IDocs on page 198.

Table 42. Miscellaneous troubleshooting items

(continued)

Area	Item
	For each configuration file, an `r3evmon` process is started to monitor the events listed. Make sure that an `r3evmon` monitoring process is started for the involved extended agent workstation.
`r3evmon:` monitoring SAP and IDoc events increases memory consumption	**Symptom**: Memory consumption increases continuously during monitoring of IDoc and standard SAP events.
	Solution: Refer to SAP Notes® 1021071 and 1109413.
`r3batch:` Duplicated events generated during IDoc monitoring	**Symptom**: The action defined in an event rule with `IDOCEventGenerated` event type is unexpectedly repeated.

	Solution: Reset the start date and time for the next monitoring loop. These values are stored in the following file:
	`<data_dir>/methods/r3evmon_cfg/XAname_r3idocmon.cfg`
	where `XAname` is the name of the SAP extended agent workstation. Therefore you can either:
	• Stop `r3evmon`, delete the *XAname*`_r3idocmon.cfg` file and then start `r3evmon` again.
	- OR -
	• Stop `r3evmon`, set the date and time in the *XAname*`_r3idocmon.cfg` file to the values you want, and start`r3evmon` again.
	Use the following format for the start date and time:
	`start_date=YYYYMMDD`
	`start_time=HHMMSS`
	For example:
	`start_date=20080307`
	`start_time=115749`
	Check the value of the **idoc_no_history** option:
	• If it is set to OFF and no *XAname*`_r3idocmon.cfg` file exists, then all matching IDocs are retrieved, not only the current ones.
	• If it is set to ON (default value), check the date and time in the *XAname*`_r3idocmon.cfg` file.

`r3batch:` No event is generated during IDoc monitoring	**Symptom**: The expected event actions are not triggered.

Table 42. Miscellaneous troubleshooting items

(continued)

Area	Item
	Solution: Check the value of the **idoc_no_history** option; if it is set to ON (default value), check the date and time in the *XAname*`_r3idocmon.cfg` file.

Error defining an internetwork dependency based on SAP event	**Symptom**: If you work with XBP 2.0, when you try to define an internetwork dependency based on an SAP event, the following error message is displayed:
	`*** ERROR 778 *** EEWO0778E An internal error has occurred. The program could not modify the following job:`
	`Job name:`
	`Job ID:`
	`%CJ ERROR`

	Solution: Perform the following steps: 1. Check if the BCTEST report is defined in your SAP system by invoking either one of the following transactions: **`sa38`** Enter BTC* and click the picklist button. In the panel that opens, click the picklist button and check if BTCTEST is shown in the list that is displayed. **`se38`** Enter BTC* and click the picklist button. Check if BTCTEST is shown in the list that is displayed. 2. If report BTCTEST is not found in the list, you can either: ◦ Choose another existing report, and, in the local options file, set the placeholder_abap_step option to the name you chose. Because the report assigned to the placeholder job is run when the corresponding event is raised, ensure that you choose a dummy report. For details about the placeholder_abap_step option, see Table 15: r3batch common configuration options on page 85. - OR - ◦ Set the placeholder_abap_step option to a custom developed ABAP code of your choice.

`r3batch:` error message when scheduling SAP jobs	**Symptom**: When creating an SAP job, the following message is displayed while trying to view the details of an ABAP's variant:
	`AWSO0101E`
	`Missing ABAP routine.`
	`J_101_REPORT_ALL_SELECTIONS please install the latest ABAP routine for Maestro!!`

Table 42. Miscellaneous troubleshooting items

(continued)

Area	Item
	Solution: This defect is caused by an error in an SAP function module. SAP describes this problem and possible solutions in the SAP Notes® 0351293 and 0116354.
`r3batch:` modify job step error	You change print parameters with the `BAPI_XBP_MODIFY_JOB_STEP` function module, and subsequently, they are incorrect. As a consequence, `r3batch` gets error 221: `MSG_CANNOT_GET_PRIARC_PARAMS: "Retrieving new print and archive parameters failed"` The problem is solved by installing SAP Note 758829.
`r3batch:` modify job step error	The BAPI_XBP_MODIFY_JOB_STEP function module always uses the name of the logged-on user as the name for the step user. In this case, when submitting a job with the `-vX` options, `r3batch` creates a job by copying all the data from the original template, except the variant name of the first step (which is provided as the option parameter). This procedure is usually referred to as the "old copy". However, when adding a step to a new job, the XBP 2.0 interface ignores the user parameter passed by `r3batch`. The problem is solved by installing SAP note 758829.

`r3batch:` does not start after installation on Windows™	**Symptom**: After installing or upgrading the SAP R/3 access method to version 8.5 on a Windows™ operating system, you try to start `r3batch` but nothing happens. The following message is displayed:
	`The application failed to initialize properly.`
	`Click on OK to terminate the application.`

	Solution: Ensure that you applied the SAP Note 684106 to install the required Microsoft™ DLLs.
`r3batch:` IBM Workload Scheduler environment variables are not resolved when specified in the task string for an R/3 batch job.	**Symptom**: When IBM Workload Scheduler environment variables are used in the task string for an R/3 batch job and the job is launched, the environment variables are not resolved. The exact string used to specify the variable is used instead.

	Solution: To leverage IBM Workload Scheduler environment variables, you must modify the access method as follows:
	1. In the *`TWA_DATA_DIR`*`/methods` directory, create a file named, r3batch.cmd (on Windows™) or r3batch.sh (on UNIX®) as required, containing the following content:
	`@echo off`
	`set METHODSPATH=%~dp0`
	`call "%METHODSPATH:"=%r3batch.exe" %*`
	2. Modify the CPU XAGENT definition from r3batch to r3batch.cmd. An example follows:
	`CPUNAME NW1`
	`  DESCRIPTION "r3batch"`
	`  OS OTHER`
	`  NODE none TCPADDR 31111`
	`  FOR MAESTRO HOST STROMBOLI ACCESS "r3batch.cmd"`

Table 42. Miscellaneous troubleshooting items

(continued)

Area	Item

	``
	    TYPE X-AGENT
	    AUTOLINK OFF
	    BEHINDFIREWALL OFF
	    FULLSTATUS OFF
	END
	``
	3. To modify the CPU access method in the Symphony® file, run JnextPlan as follows:
	``
	JnextPlan -for 0000
	``

@longlink file present in installation directory	**Symptom**: After installing IBM® Workload Scheduler on a computer with an AIX® operating system where a master domain manager is already installed, a @longlink file containing the following is present in the installation directory: ``` methods/_tools/_jvm_64/lib/desktop/icons/HighContrastInverse/48x48/mimetypes/ gnome-mime-text-x-java.png ```
	Solution: The file can be ignored. It does not present any problems for the proper functioning of the product.
Job throttling does not start	**Symptom**: When you start the job throttling feature, nothing happens and the following error message is displayed: ``` EEWOJTR0207E Error, another job throttler instance is already running against the same SAP system. Foreign job throttler registration is: Client ID="clientID", Name="TWS4APPS_JOBTHROTTLER",Host="hostname", UID "UniqueID" ```
	Cause and Solution: Possible causes are: • You are running job interception collector jobs, but the job interception and job throttling features cannot run at the same time. Choose which feature to start. For detailed information, refer to Job interception and parent-child features on page 151 and Job throttling feature on page 178. • Another job throttler instance is running against the same SAP system. You can start only one job throttler instance. • A previous job throttler instance created an exclusive lock object on the SAP system that could have become permanent. To verify it, use transaction `sm12` and query for the lock object named `TWS4APPS_JOBTHROTTLER`. If the lock object exists, and you are not running any job throttler or job interception instance, remove the lock manually and restart the job throttler.
Job throttling does not start	**Symptom**: When you start the job throttling feature, nothing happens and the following error message is displayed:

Table 42. Miscellaneous troubleshooting items

(continued)

Area	Item
	`EEWOJT0209E Error, the password format is not valid.`
	Cause and Solution: Your password is encrypted in old format. To encrypt the password with the correct encryption version, use the **enigma** or **pwdcrypt** programs. For details about how to encrypt the password, see Encrypting SAP user passwords on page 97.
Job throttling does not stop	**Symptom**: When you stop the job throttling feature, nothing happens.
	Cause and Solution: You are connected as a *TWSUser* who does not have write permission on the *XAname*`_jobthrottling_cmd.txt` file. To solve this problem, delete the *XAname*`_jobthrottling_cmd.txt` file and enter the command again. For detailed information about stopping the job throttler, refer to Step 5. Starting and stopping the job throttling feature on page 181.
Job throttling: alerts for MTEs are not generated according to the threshold values set	**Symptom**: Alerts for the MTEs created by the job throttler are generated without respecting the threshold values that are set.
	Cause and Solution: You started a new job throttler instance, which, being enabled to send data to CCMS, created the related MTEs. When you include the MTEs in your monitoring set, the threshold values are automatically set according to the existing MTE class. Nevertheless, alerts are generated without respecting these values. To solve this problem, edit the MTE properties and save them again, even if you do not change anything.
Job throttling: saving MTE properties generates an informational message	**Symptom**: When you edit and save the properties of MTEs generated by the job throttler, the following informational message is displayed: `Message does not exist.`
	Cause and Solution: In the pop-up window that displays the message, click **Continue** and close the Properties window. Your settings are saved.
The system cannot intercept jobs	**Symptom**: Although the job interception feature is active on the SAP system, the intercepted jobs are kept in scheduled state.
	Cause and Solution: The job throttler feature or the Java™ Virtual Machine used by the job throttler might still be active. On each extended agent where the job throttler was started at least once, ensure that:

Table 42. Miscellaneous troubleshooting items

(continued)

Area	Item
	1. You stopped the feature. For details, see Step 5. Starting and stopping the job throttling feature on page 181. 2. The Java™ Virtual Machine used by the job throttler was stopped by the process. To search for Java™ processes, use: **On Windows™** The Process Explorer **On UNIX®** The command `ps -ef | grep throttling` If a Java™ Virtual Machine instance related to the job throttler is found, kill it.
`access method executables: r3batch, r3event, psagent`: permission denied messages in the job log.	**Symptom**: The job log reports multiple "Permission denied" messages.
	Cause and Solution: The root cause might be that the access method executable, for example, r3batch, is submitted by the `root` user and not the `twsuser`. This creates directories and files with the wrong ownership and file permissions. Verify the ownership of the following directories and files if you are running the product on UNIX® platforms. Ensure that the `twsuser` is the owner of the files and that the user has both read and write permissions on the files, and execute permission on the directories. *TWA_DATA_DIR*/methods/traces *TWA_DATA_DIR*/methods/traces/*.log
`psagent`: misleading message displayed if the local options file has no right permissions	**Symptom**: The job log shows the following message: `EEW00439E You did not specify the required options either in the global or in the local options file.` but all the mandatory options were correctly set in the options file.
	Solution: Check that the options file has `read` and `write` permissions available to the user who is trying to launch the job.
No messages written in the job log	**Symptom**: IBM Workload Scheduler does not write any messages in the job log if the file system for tracing is full or the ljuser does not have the correct permission to write in the trace directory.

The submission of a PeopleSoft job fails	**Symptom**: The submission of a PeopleSoft job fails and the IBM Workload Scheduler job log contains a Java™ exception similar to the following:
	`Exception in thread "3194" java.lang.ExceptionInInitializerError`
	`at bea.jolt.JoltSessionAttributes.<clinit>(JoltSessionAttributes.java:183)`
	`at psft.pt8.net.JoltSessionPool.createConnection(JoltSessionPool.java:363)`
	`at psft.pt8.net.JoltSessionPool.getJoltSession(JoltSessionPool.java:220)`

Table 42. Miscellaneous troubleshooting items

(continued)

Area	Item
	Cause and Solution: The `psjoa.jar` path contains special characters. Define a path without special characters.

The submission of an Oracle job fails	**Symptom**: The submission of an Oracle job fails and the IBM Workload Scheduler job log shows the following information:
	`EEWP0017 Child MCMLJ exited normally.`
	`Exit code: 1.EEWP0027 Error - Launch job failed`

	Solution: Submitting an Oracle job might fail because there is a connection problem to the Oracle database. Verify that your Oracle naming methods are set correctly. For details about how to configure naming methods, refer to the *Oracle Net Services Administrator's Guide*.
mvsjes: RACF® authorization problem on z/OS® version 1.7	**Symptom**: An `S047` abend is returned if the `EEWSERVE` started task does not have an associated RACF® owner ID.
	Solution: In the RACF® database, associate an authorized RACF® ID with the `EEWSERVE` started task as specified in Setting RACF authorizations on z/OS.
To upgrade the SAP environment, perform the following steps:	1. Delete the TWS ABAP module. 2. Upgrade SAP. 3. Install TWS ABAP module.

Chapter 10. Scheduling jobs on IBM Workload Scheduler from SAP Solution Manager

IBM Workload Scheduler and SAP Solution Manager are integrated to allow the IBM Workload Scheduler engine to run the job scheduling tasks available from the Solution Manager user interface.

The integration is provided by the SMSE Adapter, which runs on the master domain manager. The SMSE Adapter uses the SAP Solution Manager Scheduling Enabler (SMSE) interface provided by SAP to enable *external schedulers* to run the scheduling for Solution Manager.

With this integration, when you schedule a job from the `Scheduling` panel of Solution Manager, IBM Workload Scheduler takes charge of the job scheduling, monitoring, and management tasks, as well as of job triggering and notification.

Under these conditions IBM Workload Scheduler acts as an RFC-Server with a common interface for scheduling jobs. It is identified through an RFC-Destination, registered in the SMSE. The interaction between Solution Manager and IBM Workload Scheduler is based on a PUSH mechanism implemented by the SMSE interface, whereby the master domain manager responds to requests solicited by the Solution Manager job scheduling functions.

Qualified as *external scheduler* by Solution Manager, the registered master domain managers, identified by their RFC destination names, can be called or administered from the `Process Scheduling Adapter` menu item in the Solution Manager GUI.

The jobs scheduled from Solution Manager on IBM Workload Scheduler must have been previously defined in the IBM Workload Scheduler database.

A job scheduled from the `Schedule Jobs` or `Job Documentation` panels in Solution Manager to be run by IBM Workload Scheduler, is automatically mapped in a job stream that is expressly created to include the job.

Registering the master domain manager on SAP Solution Manager

The first step to run the integration is to register the master domain manager on the SAP Solution Manager system.

To register master domain manager on the SAP Solution Manager system, you must:

1. Have established a connection based on RFC or Web Services between the master and the Solution Manager system.
2. Have the SAP JCo 3.1 Patch 3 (sapjco31P_3) libraries (`jar` files and, according to your operating system, `dll`, `so`, or `sl`) installed in the `<data_dir>/methods/smseadapter/lib` directory on the master domain manager. To download 3.1 Patch 3 (sapjco31P_3), visit the Sap Service Marketplace.

   Attention: The libraries require the Microsoft Visual C++ Redistributable Package (vcredist) installed.

3. Configure the `smseadapter.properties` file located in the `<data_dir>/methods/smseadapter/lib` directory on the master.

   The file contains a `SMSE_ADAPTER_CONNECTION_n` section that can be duplicated depending on the number of connections that you want to define. You can in fact set more connection definitions for the same master, where, for example, the following can vary:

- The SAP Solution Manager system.
- The agent that is to run the workload.
- The SAP user name.

Note: A master domain manager can have only one active connection at a time via the `smseadpter`. If the adapter finds more that one section with the `startAdapter` property set to `true` (or not set to `false`), it uses the first section of properties and ignores the others.

4. Stop and start WebSphere Application Server Liberty Base. For further information, see Application server - starting and stopping.

The following is an example of `smseadapter.properties` file:

```
[SMSE_ADAPTER_CONNECTION_1]
startAdapter =
ashost =
sysnr =
client =
sid =
user =
passwd =
lang =
destination =
setDestinationAsDefault =
jobStreamNamePrefix =
agentName =
notificationThreadCheckInterval =
adminConsoleHost =
adminConsolePort =
adminConsoleUser =
adminConsoleUserPassword =
maxRegistrationAttempts =
registrationAttemptInterval =
```

This section can be repeated as many times as needed in the `smseadapter.properties` file.

The properties are:

Table 43. Properties for the `smseadapter.properties` file.

Property	Description	Required	Notes
SMSE_ADAPTER_CONNECTION_1	This is the section header. If you have more sections the last digit should differ from one section and another. If two sections contain identical property values, only the first section read is considered, the other is ignored.	✓	

Table 43. Properties for the `smseadapter.properties` file.

(continued)

Property	Description	Required	Notes
startAdapter	Specifies whether to connect or not to SAP Solution Manager. Can be true or false. Must be set to true to make the connection work. Set to false to temporarily suspend the connection.	✓	The default is true.
ashost	The host name of the SAP Solution Manager server on which the master domain manager registers. For example, / H/7.142.153.8/H/7.142.154.114.	✓	The master domain manager can connect to one Solution Manager system at a time.
sysnr	The SAP system number of the system that the master registers on. This value must have two digits. For example, 00.	✓	
client	The SAP client number. For example, 001.	✓	
sid	The SAP system identifier (SID) that the master registers on. For example, SM1.	✓	
user	The SAP user name that will be used during the notification process to log into SAP Solution Manager. For example, twsadmin.	✓	
passwd	The SAP password that will be used during the notification process to log into SAP Solution Manager. You can enter it in clear or in encrypted forms.	✓	To encrypt the password use the `enigma` program located in the `methods` folder on the master.
lang	The SAP logon language. For example, EN.	✓	
destination	A name entered here to identify the RFC Destination that will be used to connect to SAP Solution Manager. For example, IWSM2.	✓	This name defines the logical connection between the Solution Manager system and the master domain manager, referred to in Solution Manager as the `external`

Table 43. Properties for the `smseadapter.properties` file.

(continued)

Property	Description	Required	Notes
	Note: The destination name must be univocal.		`scheduler`. The complete destination name will then be formed by: *destination@mdm_name* For example: IWSM2@MAS93WIN **Note:** The destination name must be univocal.
setDestinationAsDefault	Set to `true` to make this destination the default one. The default is `false`.		Use this property in a context where a Solution Manager system has more than one active destination defined (that is, more registered masters), to set the default external scheduler. If you do not set a default, and you have more external schedulers registered on an SM system, you will have to specify the destination at scheduling time.
jobStreamNamePrefix	A prefix of at least four letters that is to be added to the names of the job streams created when jobs are submitted. The first character must be a letter while the remaining characters can be alphanumeric.		The default prefix is SOLMAN.
agentName	The name of the IWS agent that will run the jobs. When you search for the job definition in the Scheduling dialog, the Search utility returns the names of the jobs defined to run on this agent.		If no agent name is specified, the Search utility returns the names of the jobs defined to run on all the agents attached to the master domain manager (unless you use filtering).
notificationThreadCheckInterval	The time interval, in seconds, between checks made by the notification thread on the status changes of a job. The default is 5 seconds.		The thread notifies Solution Manager with the status changes of a job.
adminConsoleURL	The protocol used (http or https) and the host name and port of the Dynamic Workload Console attached to the master. For example, `https://mydwc:port_number/abc/console`.		The next four properties, all related to the Dynamic Workload Console, are optional, but if you specify one, you must specify all.

Table 43. Properties for the `smseadapter.properties` file.

(continued)

Property	Description	Required	Notes
adminConsoleUser	The username that logs onto the Dynamic Workload Console attached to the master.		
adminConsoleUserPassword	The password of the username that logs onto the Dynamic Workload Console attached to the master.		
maxRegistrationAttempts	The maximum number of attempts to connect to SAP. By default, it is set to 5 times.		
registrationAttemptInterval	The time after which a new attempt to connect to SAP is performed. By default, it is set to 5 seconds.		

Note: If the language configured for the master domain manager is different from the language configured for the Solution Manager system, the messages issued in the Solution Manager user interface may be displayed in mixed languages.

Scheduling

The `Job Management Administration` panel of Solution Manager has two entry points for scheduling jobs:

- The `Schedule Jobs` item in `Common Tasks`, a direct way of scheduling, where you pick the job from the IBM Workload Scheduler database and you set the scheduling options and time definitions.
- The `Job Documentation` object, where you can create and edit job documentation, schedule, monitor, and manage jobs.

The jobs scheduled from Solution Manager on IBM Workload Scheduler must have been previously defined in the IBM Workload Scheduler database.

A job scheduled from the `Schedule Jobs` or `Job Documentation` panels in Solution Manager to be run by IBM Workload Scheduler, is automatically mapped in a job stream that is expressly created to include the job. The job stream is (automatically) defined in the IBM Workload Scheduler database with a specific prefix defined in the smseadapter.properties file on page 236.

Scheduling jobs directly

In the Scheduling panel, before you can proceed to enter the job name and the scheduling details, you are asked to specify the identity of the scheduling system and the scheduler type, which must be SMSE. You can then specify the name of the job definition, and the job type, which can be any of the job types supported by IBM Workload Scheduler. The job is qualified by Solution Manager as an `external` job.

Select the `Status message` check box to enable monitoring tasks for the job.

In the `Start Conditions` section select when and how frequently the job will run and optionally make selections in the `Repeat every` and `Time Period` groups. Your selections are then mapped to matching run cycles, `valid from` and `valid to` dates, working days or non- working days, and time dependencies on IBM Workload Scheduler.

Note: The Extended Window start condition is not supported. All other start conditions are supported.

Use the tabs available on the uppermost part of the panel to manage jobs; for example, to copy, reschedule, release, kill, or cancel a job, create and see external notes, and check external logs.

If a scheduled job has not been started, you can change its start conditions or parameters and click **Schedule/Change Externally** again. Alternatively, you can change the start conditions and select **Reschedule** to reset the job to a new start time. In either case, IBM Workload Scheduler deletes the current job instance (that has not been started) and creates another one with the new characteristics.

On the other hand, you can click **Cancel** on a non-completed job that was already started. In this case, IBM Workload Scheduler deletes the running instance as expected.

As soon as the job is scheduled with success, the external job ID and the status are updated and you can view the job instance on the Dynamic Workload Console.

Scheduling from job documentation

With the Job Documentation option of the Job Management Administration panel, you can also create job documentation for jobs defined in IBM Workload Scheduler and scheduled from Solution Manager. From the Job Documentation menu you can view and edit job details, including job steps, basic business information, and scheduling information.

To create job documentation:

1. In the Job Documentation view, create job documentation **for a Job with detail UI**.
2. In the General pane of the new job documentation creation page, enter a job documentation name and select **Other** as *Job Type*. Selecting **Other**, *External Scheduler* is automatically selected.
3. Add a step in the Step Overview table.
4. Select a job definition type from a list of job types available from IBM Workload Scheduler.
5. Click **Save** on top of the job documentation creation page.
6. Select the **Systems** tab in the job documentation creation page and add a solution documentation, a logical component group, or a technical scenario for the new job documentation in the *Logical Component Groups, Solution Documentation and Technical Scenarios* table. Click **Save**.
7. Select **Scheduling** in the Systems table to set up scheduling definitions for the job associated with the new job documentation.

   This action displays the same Scheduling panel described in Scheduling jobs directly on page 237.

   You can also select **Configure Monitoring** in the Systems table to set up monitoring specifications for the job.

Monitoring

Job status retrieval and job monitoring tasks are run by IBM Workload Scheduler, but you can configure and view them from the Solution Manager `Job Documentation` and `Monitoring` views. In Solution Manager to monitor a job you must configure a Business project monitoring object (BPmon). When monitoring data is requested for a job, Solution Manager through the SMSE adapter requests IBM Workload Scheduler for updates on the job status, logs, alerts.

To view the status of a job in the Solution Manager `Job Documentation` view, provided you selected the `Status message` check box in the `Scheduling` page, follow these steps:

1. Open the job documentation view for the job.
2. Select the `Systems` and the `Scheduling` tabs.
3. In the `Scheduling` page select the `External Log` button.

   The job log is displayed in a pop-up window.
4. Select the `Refresh` button of the `External Job Status` field in the `Scheduling` page.

   The current status of the job is displayed in the field.

To configure monitoring for a scheduled job with the `Status message` check box selected, go to the `Job Management Administration` panel of Solution Manager and open the `Job Documentation` view related to the job. There, select the `Systems` tab and in the ensuing page select `Configure Monitoring`.

1. In the `Identification` section of the `Job Monitoring Setup` window, input all mandatory fields.
2. Select the `Alert Setting and Data Collection` tab and configure alerts to your convenience.
3. Fill in the mandatory fields in the `Incidents and Notifications` and `Monitoring Activities` tabs.
4. Select the `Mass Generate and Activate` button on top to save and activate the monitoring object.

With the Push mechanism IBM Workload Scheduler forwards to Solution Manager the status changes that a job instance undergoes until it reaches a final status such as `complete`, `canceled`, `error`, or `killed`. IBM Workload Scheduler also forwards the following scheduling time information for the job instance:

- Estimated duration
- Actual start
- Actual completion

On the basis of this information, and according to the alert configuration you specified in the `Alert Setting and Data Collection` pane, Solution Manager triggers these alerts when any of the thresholds you specified are reached or exceeded. This grants you the means to keep the execution of your workload under control.

To view the alerts for a monitored job, select the `Unified Alert Inbox` view in the `Job Management Administration` panel:

1. Select the monitoring object for the job in the Job Monitoring Standard View.
2. Refresh the alert list table after some monitoring period.

Note: `Alert Inbox` and `Job Monitoring` panels can be also accessed from the `Home` in the `Job Management` section.

Setting the traces on the application server for the major IBM Workload Scheduler processes

About this task

The application server handles all communications between the IBM Workload Scheduler processes. The trace for these communications is set to `tws_info` by default (information messages only). The application server can be set to trace `all` communications, either for the whole product or for these specific groups of processes:

- Command line
- Connector
- Database
- Planner
- Utilities
- Dynamic workload broker

Significant impact on performance: Activating traces for the WebSphere Application Server Liberty Base leads to a significant impact on performance, especially if you set the tracing to `all`. Thus you are strongly advised to identify the process group where the problem that you want to trace is occurring, and only set the trace to that group.

To modify the trace level on the WebSphere Application Server Liberty Base, edit the `trace.xml` file as necessary.

Templates for the master domain manager are stored in the following paths:

UNIX **On UNIX operating systems**

TWA_home`/usr/servers/engineServer/configDropins/templates`

Windows **On Windows operating systems**

TWA_home`\usr\servers\engineServer\configDropins\templates`

Templates for the Dynamic Workload Console are stored in the following paths:

UNIX **On UNIX operating systems**

DWC_home`/usr/servers/dwcServer/configDropins/templates`

Windows **On Windows operating systems**

DWC_home`\usr\servers\dwcServer\configDropins\templates`

When you edit the file with your customized settings for the master domain manager, move it to the following paths:

UNIX **On UNIX operating systems**

TWA_DATA_DIR`/usr/servers/engineServer/configDropins/overrides`

On Windows operating systems

TWA_home\usr\servers\engineServer\configDropins\overrides

When you edit the file with your customized settings for the Dynamic Workload Console, move it to the following paths:

On UNIX operating systems

DWC_DATA_dir/usr/servers/dwcServer/configDropins/overrides

On Windows operating systems

DWC_home\usr\servers\dwcServer\configDropins\overrides

1. Copy the template file from the `templates` folder to a working folder.
2. Edit the template file in the working folder with the desired configuration.
3. Optionally, create a backup copy of the relevant configuration file present in the `overrides` directory in a different directory. Ensure you do not copy the backup file in the path where the template files are located.
4. Copy the updated template file to the `overrides` folder. Maintaining the original folder structure is not required.
5. Changes are effective immediately.

For example, to modify the trace level on WebSphere Application Server Liberty Base, perform the following steps:

1. Copy the `trace.xml` file from the TWA_home/usr/servers/engineServer/configDropins/templates folder to a working folder.
2. Edit the template file in the working folder by changing the following string:

   ```
   <variable name="trace.specification" value="*info"/>
   ```

   into

   ```
   <variable name="trace.specification" value="com.ibm.tws.dao.model.
   =all:com.ibm.tws.dao.rdbms.=all"/>
   ```

3. Copy the updated template file to *TWA_DATA_DIR* /usr/servers/engineServer/configDropins/overrides. Changes are effective immediately.

Traces are stored in *DWC_DATA_dir*/appserver/engineServer/logs.

The **trace.specification** can be found in `trace.xml` files and can refer to a specific component (tws*_xxx*) or to the whole product, as follows:

tws_all

"com.ibm.tws.**=all:org.apache.wink.server.**=all:com.hcl.tws.*=all"

tws_alldefault

"com.ibm.tws.*=error=enabled"

tws_broker_all

"com.ibm.scheduling.*=all:TWSAgent=all"

tws_rest

" com.ibm.tws.conn.**=all:com.ibm.tws.twsd.rest.**=all:org.apache.wink.server.*=all"

tws_cli

"com.ibm.tws.cli.**=all:com.ibm.tws.objects.**=all"

tws_utils

"com.ibm.tws.util.*=all"

tws_conn

"com.ibm.tws.conn.**=all:com.ibm.tws.objects.**=all:com.ibm.tws.updatemanager.**=all:com.ibm.tws.dao.plan.**=all"

tws_db

"com.ibm.tws.dao.model.**=all:com.ibm.tws.dao.rdbms.**=all"

tws_planner

"com.ibm.tws.planner.**=all:com.tivoli.icalendar.**=all:com.ibm.tws.runcycles.**=all:com.ibm.tws.conn.planner.**=all:com.ibm.tws.cli.planner.*=all"

tws_secjni

"com.ibm.tws.audit.**=all:com.ibm.tws.security.**=all"

tws_engine_broker_all

"com.ibm.tws.**=all:com.ibm.scheduling.**=all:TWSAgent=all"

Editing the logging element above with the **traceSpecifcation** value to `tws_all`, enables "**com.ibm.tws.=all:org.apache.wink.server.=all:com.hcl.tws.*=all**".

Other values are reported in variable tags above. You can also replace the value of the **trace.specification** parameter with a custom string.

Notices

This document provides information about copyright, trademarks, terms and conditions for product documentation.

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.

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.

Notices and information

The IBM license agreement and any applicable notices on the web download page for this product refers You to this file for details concerning terms and conditions applicable to software code identified as excluded components in the License Information document and included in IBM Workload Scheduler for Applications 8.4.0 (the "Program").

Notwithstanding the terms and conditions of any other agreement you may have with IBM or any of its related or affiliated entities (collectively "IBM"), the third party software code identified below are "Excluded Components" and are subject to the terms and conditions of the License Information document accompanying the Program and not the license terms that may be contained in the notices below. The notices are provided for informational purposes.

The Program includes the following Excluded Components:

- libmsg
- Jakarta ORO
- ISMP Installer

- HSQLDB
- Quick
- Infozip

Libmsg

For the code entitled Libmsg

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that Alfalfa's name not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

ALPHALPHA DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL ALPHALPHA BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Apache Jakarta ORO

For the code entitled Jakarta ORO

The Apache Software License, Version 1.1

Copyright (c) 2000-2002 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
4. The names "Apache" and "Apache Software Foundation", "Jakarta-Oro" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
5. Products derived from this software may not be called "Apache" or "Jakarta-Oro", nor may "Apache" or "Jakarta-Oro" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see http://www.apache.org/.

Portions of this software are based upon software originally written by Daniel F. Savarese. We appreciate his contributions.

ISMP Installer (InstallShield 10.50x)

For the code entitled ISMP Installer (InstallShield 10.50x)

The Program includes the following Excluded Components:

- Quick V1.0.1
- HSQLDB V1.7.1
- InfoZip Unzip stub file V5.40, V5.41,V5.42 & V5.5

JXML CODE

For the code entitled Quick

JXML CODE. The Program is accompanied by the following JXML software:

- Quick V1.0.1

IBM is required to provide you, as the recipient of such software, with a copy of the following license from JXML:

```
Copyright (c) 1998, 1999, JXML, Inc.
All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

All product materials mentioning features or use of this software must display the following acknowledgement:

- This product includes software developed by JXML, Inc. and its contributors: http://www.jxml.com/mdsax/contributers.html

Neither name of JXML nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY JXML, INC. AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JXML OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

InfoZip CODE

For the code entitled InfoZip

InfoZip CODE. The Program is accompanied by the following InfoZip software:

- One or more of: InfoZip Unzipsfx stub file V5.40, V5.41, V5.42 & V5.5

IBM is required to provide you, as the recipient of such software, with a copy of the following license from InfoZip:

- This is version 2000-Apr-09 of the Info-ZIP copyright and license.

  The definitive version of this document should be available at ftp://ftp.info-zip.org/pub/infozip/license.html indefinitely.

Copyright (c) 1990-2000 Info-ZIP. All rights reserved.

For the purposes of this copyright and license, "Info-ZIP" is defined as the following set of individuals:

- Mark Adler, John Bush, Karl Davis, Harald Denker, Jean-Michel Dubois, Jean-loup Gailly, Hunter Goatley, Ian Gorman, Chris Herborth, Dirk Haase, Greg Hartwig, Robert Heath, Jonathan Hudson, Paul Kienitz, David Kirschbaum, Johnny Lee, Onno van der Linden, Igor Mandrichenko, Steve P. Miller, Sergio Monesi, Keith Owens, George Petrov, Greg Roelofs, Kai Uwe Rommel, Steve Salisbury, Dave Smith, Christian Spieler, Antoine Verheijen, Paul von Behren, Rich Wales, Mike White

This software is provided "as is," without warranty of any kind, express or implied. In no event shall Info-ZIP or its contributors be held liable for any direct, indirect, incidental, special or consequential damages arising out of the use of or inability to use this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. Redistributions of source code must retain the above copyright notice, definition, disclaimer, and this list of conditions.
2. Redistributions in binary form must reproduce the above copyright notice, definition, disclaimer, and this list of conditions in documentation and/or other materials provided with the distribution.
3. Altered versions--including, but not limited to, ports to new operating systems, existing ports with new graphical interfaces, and dynamic, shared, or static library versions--must be plainly marked as such and must not be misrepresented as being the original source. Such altered versions also must not be misrepresented as being Info-ZIP releases--including, but not limited to, labeling of the altered versions with the names "Info-ZIP" (or any variation thereof, including, but not limited to, different capitalizations), "Pocket UnZip," "WiZ" or "MacZip" without the explicit permission of Info-ZIP. Such altered versions are further prohibited from misrepresentative use of the Zip-Bugs or Info-ZIP e-mail addresses or of the Info-ZIP URL(s).
4. Info-ZIP retains the right to use the names "Info-ZIP," "Zip," "UnZip," "WiZ," "Pocket UnZip," "Pocket Zip," and "MacZip" for its own source and binary releases.

HSQL Code

For the code entitled HSQLDB

HSQL CODE. The Program is accompanied by the following HSQL Development Group software:

- HSQLDB V1.7.1

IBM is required to provide you, as the recipient of such software, with a copy of the following license from the HSQL Development Group:

Copyright (c) 2001-2002, The HSQL Development Group All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the HSQL Development Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQLDB.ORG, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN

CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

HP-UX Runtime Environment, for the Java 2 Platform

TERMS FOR SEPARATELY LICENSED CODE

This Program includes HP Runtime Environment for J2SE HP-UX 11i platform software as a third party component, which is licensed to you under the terms of the following HP-UX license agreement and not those of this Agreement

HP-UX Runtime Environment license text

HP-UX Runtime Environment, for the Java 2 Platform

ATTENTION: USE OF THE SOFTWARE IS SUBJECT TO THE HP SOFTWARE LICENSE TERMS AND SUPPLEMENTAL RESTRICTIONS SET FORTH BELOW, THIRD PARTY SOFTWARE LICENSE TERMS FOUND IN THE THIRDPARTYLICENSEREADME.TXT FILE AND THE WARRANTY DISCLAIMER ATTACHED. IF YOU DO NOT ACCEPT THESE TERMS FULLY, YOU MAY NOT INSTALL OR OTHERWISE USE THE SOFTWARE. NOTWITHSTANDING ANYTHING TO THE CONTRARY IN THIS NOTICE, INSTALLING OR OTHERWISE USING THE SOFTWARE INDICATES YOUR ACCEPTANCE OF THESE LICENSE TERMS.

HP SOFTWARE LICENSE TERMS

The following terms govern your use of the Software unless you have a separate written agreement with HP. HP has the right to change these terms and conditions at any time, with or without notice.

License Grant

HP grants you a license to Use one copy of the Software. "Use" means storing, loading, installing, executing or displaying the Software. You may not modify the Software or disable any licensing or control features of the Software. If the Software is licensed for "concurrent use", you may not allow more than the maximum number of authorized users to Use the Software concurrently.

Ownership

The Software is owned and copyrighted by HP or its third party suppliers. Your license confers no title or ownership in the Software and is not a sale of any rights in the Software. HP's third party suppliers may protect their rights in the event of any violation of these License Terms.

Third Party Code

Some third-party code embedded or bundled with the Software is licensed to you under different terms and conditions as set forth in the THIRDPARTYLICENSEREADME.txt file. In addition to any terms and conditions of any third party license identified in the THIRDPARTYLICENSEREADME.txt file, the disclaimer of warranty and limitation of liability provisions in this license shall apply to all code distributed as part of or bundled with the Software.

Source Code

Software may contain source code that, unless expressly licensed for other purposes, is provided solely for reference purposes pursuant to the terms of this license. Source code may not be redistributed unless expressly provided for in these License Terms.

Copies and Adaptations

You may only make copies or adaptations of the Software for archival purposes or when copying or adaptation is an essential step in the authorized Use of the Software. You must reproduce all copyright notices in the original Software on all copies or adaptations. You may not copy the Software onto any bulletin board or similar system.

No Disassembly or Decryption

You may not disassemble or decompile the Software unless HP's prior written consent is obtained. In some jurisdictions, HP's consent may not be required for disassembly or decompilation. Upon request, you will provide HP with reasonably detailed information regarding any disassembly or decompilation. You may not decrypt the Software unless decryption is a necessary part of the operation of the Software.

Transfer

Your license will automatically terminate upon any transfer of the Software. Upon transfer, you must deliver the Software, including any copies and related documentation, to the transferee. The transferee must accept these License Terms as a condition to the transfer.

Termination

HP may terminate your license upon notice for failure to comply with any of these License Terms. Upon termination, you must immediately destroy the Software, together with all copies, adaptations and merged portions in any form.

Export Requirements

You may not export or re-export the Software or any copy or adaptation in violation of any applicable laws or regulations.

This software or any copy or adaptation may not be exported, reexported or transferred to or within countries under U.S. economic embargo including the following countries: Afghanistan (Taliban-controlled areas), Cuba, Iran, Iraq, Libya, North Korea, Serbia, Sudan and Syria. This list is subject to change.

This software or any copy or adaptation may not be exported, reexported or transferred to persons or entities listed on the U.S. Department of Commerce Denied Parties List or on any U.S. Treasury Department Designated Nationals exclusion list, or to any party directly or indirectly involved in the development or production of nuclear, chemical, biological weapons or related missile technology programs as specified in the U.S. Export Administration Regulations (15 CFR 730).

U.S. Government Contracts

If the Software is licensed for use in the performance of a U.S. government prime contract or subcontract, you agree that, consistent with FAR 12.211 and 12.212, commercial computer Software, computer Software documentation and technical data for commercial items are licensed under HP's standard commercial license.

SUPPLEMENTAL RESTRICTIONS

You acknowledge the Software is not designed or intended for use in on-line control of aircraft, air traffic, aircraft navigation, or aircraft communications; or in the design, construction, operation or maintenance of any nuclear facility. HP disclaims any express or implied warranty of fitness for such uses.

ADDITIONAL SUPPLEMENTAL RESTRICTIONS FOR HP-UX RUNTIME ENVIRONMENT, FOR THE JAVA(TM) 2 PLATFORM

- * License to Distribute HP-UX Runtime Environment, for the Java(tm) 2 Platform. You are granted a royalty-free right to reproduce and distribute the HP-UX Runtime Environment, for Java provided that you distribute the HP-UX Runtime Environment, for the Java 2 Platform complete and unmodified, only as a part of, and for the sole purpose of running your Java compatible applet or application ("Program") into which the HP-UX Runtime Environment, for the Java 2 Platform is incorporated.
- * Java Platform Interface. Licensee may not modify the Java Platform Interface ("JPI", identified as classes contained within the "java" package or any subpackages of the "java" package), by creating additional classes within the JPI or otherwise causing the addition to or modification of the classes in the JPI. In the event that Licensee creates any Java-related API and distributes such API to others for applet or application development, Licensee must promptly publish broadly, an accurate specification for such API for free use by all developers of Java-based software.
- * You may make the HP-UX Runtime Environment, for the Java 2 Platform accessible to application programs developed by you provided that the programs allow such access only through the Invocation Interface specified and provided that you shall not expose or document other interfaces that permit access to such HP-UX Runtime Environment, for the Java 2 Platform. You shall not be restricted hereunder from exposing or documenting interfaces to software components that use or access the HP-UX Runtime Environment, for the Java 2 Platform.

HP WARRANTY STATEMENT

DURATION OF LIMITED WARRANTY: 90 DAYS

HP warrants to you, the end customer, that HP hardware, accessories, and supplies will be free from defects in materials and workmanship after the date of purchase for the period specified above. If HP receives notice of such defects during the warranty period, HP will, at its option, either repair or replace products which prove to be defective. Replacement products may be either new or equivalent in performance to new.

HP warrants to you that HP Software will not fail to execute its programming instructions after the date of purchase, for the period specified above, due to defects in materials and workmanship when properly installed and used. If HP receives notice of such defects during the warranty period, HP will replace Software which does not execute its programming instructions due to such defects.

HP does not warrant that the operation of HP products will be uninterrupted or error free. If HP is unable, within a reasonable time, to repair or replace any product to a condition warranted, you will be entitled to a refund of the purchase price upon prompt return of the product. Alternatively, in the case of HP Software, you will be entitled to a refund of the purchase price upon prompt delivery to HP of written notice from you confirming destruction of the HP Software, together with all copies, adaptations, and merged portions in any form.

HP products may contain remanufactured parts equivalent to new in performance or may have been subject to incidental use.

Warranty does not apply to defects resulting from: (a) improper or inadequate maintenance or calibration; (b) software, interfacing, parts or supplies not supplied by HP, (c) unauthorized modification or misuse; (d) operation outside of the published environmental specifications for the product, (e) improper site preparation or maintenance, or (f) the presence of code from HP suppliers embedded in or bundled with any HP product.

TO THE EXTENT ALLOWED BY LOCAL LAW, THE ABOVE WARRANTIES ARE EXCLUSIVE AND NO OTHER WARRANTY OR CONDITION, WHETHER WRITTEN OR ORAL, IS EXPRESSED OR IMPLIED AND HP SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, SATISFACTORY QUALITY, AND FITNESS FOR A PARTICULAR PURPOSE. Some countries, states, or provinces do not allow limitations on the duration of an implied warranty, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights and you might also have other rights that vary from country to country, state to state, or province to province.

TO THE EXTENT ALLOWED BY LOCAL LAW, THE REMEDIES IN THIS WARRANTY STATEMENT ARE YOUR SOLE AND EXCLUSIVE REMEDIES. EXCEPT AS INDICATED ABOVE, IN NO EVENT WILL HP OR ITS SUPPLIERS BE LIABLE FOR LOSS OF DATA OR FOR DIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL (INCLUDING LOST PROFIT OR DATA), OR OTHER DAMAGE, WHETHER BASED IN CONTRACT, TORT, OR OTHERWISE. Some countries, states, or provinces do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation may not apply to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Index

S

W