

IBM Workload Scheduler
Administration
Version 9.5 Fix Pack 7

Note

Before using this information and the product it supports, read the information in [Notices on page cdxciiv](#).

This edition applies to version 9, release 5, modification level 0 of IBM Workload Scheduler (program number 5698-WSH) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

List of Figures.....	viii	Archiving job data.....	105
List of Tables.....	ix	Configuring to schedule J2EE jobs.....	106
About this publication.....	xi	Configuring to schedule job types with advanced options.....	113
What is new in this release.....	xi	Configuring security roles for users and groups.....	114
Accessibility	xi	Configuring command-line client access authentication.....	117
Technical training.....	xi	Connection parameters.....	117
Support information.....	xi	Entering passwords.....	120
Chapter 1. Customizing and configuring IBM Workload Scheduler.....	13	An active-active high availability scenario.....	120
Personalizing UI labels.....	13	IBM Workload Scheduler console messages and prompts.....	125
Setting global options.....	14	Setting sysloglocal on UNIX™.....	125
Global options - summary.....	16	console command.....	126
Global options - detailed description.....	29	Modifying jobmon service rights for Windows™.....	126
Setting local options.....	51	Chapter 2. Configuring the Dynamic Workload Console.....	127
Localopts summary.....	51	Launching in context with the Dynamic Workload Console.....	127
Localopts details.....	55	Scenarios.....	127
Local options file example.....	72	Advanced optional parameters	128
Setting user options.....	76	Configuring access to the Dynamic Workload Console.....	136
Sample useropts file.....	76	Configuring a user registry.....	137
Multiple product instances.....	77	Configuring roles to access the Dynamic Workload Console.....	137
Configuring the agent.....	77	Configuring the Dynamic Workload Console for Single Sign-On.....	140
Configuring general properties [ITA].....	80	How to configure the Dynamic Workload Console and the master domain manager for Single Sign-On.....	141
Configuring log message properties [JobManager.Logging.clog].....	81	How to configure the Dynamic Workload Console 9.5 and a master domain manager 9.4.x for Single Sign-On.....	142
Configuring trace properties when the agent is stopped [JobManager.Logging.clog].....	82	Configuring Dynamic Workload Console to use SSL.....	145
Trace configuration for the agent.....	84	Customizing your global settings.....	145
Configuring common launchers properties [Launchers].....	87	Customize video URLs.....	148
Configuring properties of the native job launcher [NativeJobLauncher].....	89	Override graphical view limits.....	149
Configuring properties of the Java™ job launcher [JavaJobLauncher].....	92	Plan View in new window.....	149
Configuring properties of the Resource advisor agent [ResourceAdvisorAgent].....	92	Plan View auto refresh interval.....	150
Configuring properties of the System scanner [SystemScanner].....	95	Disable and customize NewsFeed function.....	150
Configuring environment variables [Env].....	95	Disable and customize the creation of predefined tasks.....	152
Regular agent maintenance.....	95	Add customized URL to job and job streams.....	153
Configuring the dynamic workload broker server on the master domain manager and dynamic domain manager.....	96	User registry.....	155
Maintaining the dynamic workload broker server on the master domain manager and dynamic domain manager.....	98	z/OS http connections.....	156
Enabling unsecure communication with the dynamic workload broker server.....	98	Limit the number of objects retrieved by queries.....	156
ResourceAdvisorConfig.properties file.....	99	Limit task and engine sharing.....	157
JobDispatcherConfig.properties file.....	101		
BrokerWorkstation.properties file	104		

Show all dependencies.....	158	Network operation.....	273
Auditing mobile app activity.....	159	Network processes.....	274
Modifying the number of archived plans displayed in the Dynamic Workload Console.....	160	Optimizing the network.....	278
Show or hide predecessors from What-if Analysis Gantt view.....	160	Data volumes.....	278
TdwcGlobalSettings.xml sample.....	160	Connectivity.....	279
Disable the What-if Analysis	165	Planning space for queues.....	280
Configuring High Availability.....	165	Tuning mailman servers.....	289
Configuring Dynamic Workload Console to view reports.....	166	Netman configuration file.....	290
Configuring for an Oracle database.....	166	Determining internal Symphony table size.....	291
Chapter 3. Configuring user authorization (Security file).....	168	Defining access methods for agents.....	292
Getting started with security	168	UNIX™ access methods.....	293
Role-based security model.....	169	IP address validation.....	296
Configuring role-based security from Dynamic Workload Console.....	170	Support for Internet Protocol version 6.....	296
Configuring role-based security with composer command-line.....	176	Operating system configuration (UNIX™ only)....	297
Actions on security objects.....	185	IP address validation messages.....	297
Attributes for object types.....	191	Impact of network changes.....	298
Specifying object attribute values.....	192	Chapter 6. Connection security overview.....	300
Classic security model.....	196	Creating a Certificate Authority.....	301
Security management overview.....	196	SSL connection by using the default certificates.....	302
Updating the security file.....	197	Customizing certificates for master domain manager and dynamic agent communication.....	303
Centralized security management.....	201	Customizing master domain manager and dynamic agent certificates.....	304
Configuring the security file.....	202	Customizing master domain manager certificates.....	305
Sample security file.....	242	Customizing dynamic agent certificates.....	307
Chapter 4. Configuring authentication.....	252	Scenario: Connection between the Dynamic Workload Console and the IBM Workload Scheduler components.....	308
Where to configure authentication.....	252	Overview.....	309
Available configurations.....	252	Customizing certificates for master domain manager and Dynamic Workload Console communication.....	311
Rules for using a Federated User Registry with IBM Workload Scheduler.....	253	Extending communication scenarios to other server components.....	315
Completing the LDAP configuration.....	253	Creating new brand keystores.....	316
Configuring an IBM Tivoli Directory Server.....	255	Creating brand new keystores for the server components.....	316
Example configurations of LDAP servers for IDS.....	256	Creating brand new keystores for the dynamic agent.....	316
Configuring Microsoft Active Directory.....	258	Scenario: SSL Communication across the fault-tolerant agent network.....	317
Example configurations of LDAP servers for Microsoft Active Directory.....	259	Using SSL for netman and conman.....	317
Configuring an OpenID Connect Client.....	262	SSL command reference.....	327
Chapter 5. Network administration.....	264	twManageKey script for keystore conversion.....	329
Network overview.....	264	FIPS compliance.....	330
Network definition.....	265	FIPS overview.....	331
Network communications.....	266	Using FIPS certificates.....	331
Network links.....	266	Configuring SSL to be FIPS-compliant.....	336
Working across firewalls.....	267	Finding the GSKit version on agents running on UNIX™ and Linux™ operating systems.....	339
Configuring dynamic agent communications through a gateway.....	268		
Enabling Ports.....	272		

Chapter 7. Data maintenance.....	340	Changing key IBM Workload Scheduler passwords.....	412
Maintaining the database.....	340	Changing the WebSphere Application Server Liberty Base user ID and password.....	413
Backing up and restoring.....	340	Change password used by command-line clients to access the master domain manager.....	415
Reorganizing the database.....	342	Change password used by fault-tolerant agent systems to access the master domain manager (for conman).....	415
Maintaining the file system.....	343	Update the engine connection parameters in the GUIs.....	416
Avoiding full file systems.....	343	Windows™ - update Windows™ services.....	416
Log files and archived files.....	349	Change the IBM Workload Scheduler user definition.....	416
Temporary files.....	355	Unlinking and stopping IBM Workload Scheduler.....	417
Managing event message queue file sizes.....	355	Changing the properties for the database.....	418
Administrative tasks - Databases.....	356	Changing the workstation host name or IP address.....	420
Administrative tasks - DB2®.....	356	Reporting the changes in the WebSphere Application Server Liberty Base configuration file.....	421
Changing DB2® passwords.....	357	Reporting the changed host name or IP address in the workstation definition.....	422
Locating the DB2® tools.....	357	Reporting the changed host name or IP address of the dynamic workload broker server.....	423
User permissions for running the DB2® tools....	357	Reporting the changed host name or IP address of the dynamic agent.....	424
Running DB2® maintenance manually.....	358	Changing the security settings.....	424
Reorganizing the DB2® database.....	359	Managing the event processor.....	426
Monitoring the lock list memory.....	360	Automatically initializing IBM Workload Scheduler instances.....	426
Administrative tasks - Oracle.....	362	Configuring IBM Workload Scheduler using templates.....	428
Changing the Oracle access password.....	362	WebSphere Application Server Liberty Base tasks....	432
Maintaining the Oracle database.....	363	Application server - starting and stopping	432
Obtaining information about the IBM Workload Scheduler databases installed on an Oracle instance.....	363	Application server - automatic restart after failure.....	434
User permissions for running the Oracle tools.....	363	Application server - encrypting the profile properties files.....	437
Modifying your RDBMS server	363	Application server - configuration files backup and restore.....	437
Maintaining audit trails.....	364	Application server - changing the host name or TCP/IP ports.....	437
Database and plan audit.....	365	Application server - changing the trace properties.....	438
Dynamic workload scheduling audit.....	373	Chapter 9. Administering IBM i dynamic environment... 440	
Keeping track of database changes using audit reports.....	383	Configuring the agent on IBM i systems.....	440
Collecting job metrics.....	388	Configuring log message properties [JobManager.Logging.ccllog].....	441
Job metrics queries for DB2.....	388	Configuring trace properties when the agent is stopped [JobManager.Logging.ccllog].....	442
Job metrics queries for DB2 for zOS.....	389	Trace configuration for the agent.....	443
Job metrics queries for Oracle database.....	389	Configuring common launchers properties [Launchers].....	447
Chapter 8. Administrative tasks.....	390		
Switching a domain manager.....	391		
Simplified procedure for switching a domain manager.....	392		
Complete procedure for switching a domain manager.....	393		
Switching the master to a backup.....	396		
Selecting a workstation for the backup master domain manager.....	397		
Automatic failover.....	399		
Manually switching the master.....	405		
Switching a dynamic domain manager for a Z controller.....	408		
Cloning scheduling definitions from one environment to another.....	409		

Configuring properties of the native job launcher [NativeJobLauncher].....	449
Configuring properties of the Java™ job launcher [JavaJobLauncher].....	452
Configuring properties of the Resource advisor agent [ResourceAdvisorAgent].....	452
Configuring properties of the System scanner [SystemScanner].....	454
Configuring to schedule job types with advanced options.....	455
Customizing the SSL connection between a master domain manager or a dynamic domain manager and IBM i agents connected to it using your own certificates.....	456
Chapter 10. Performance.....	463
Network traffic.....	463
Tracing.....	463
Logging.....	463
Maintaining the database.....	463
Symphony file sizing.....	463
Tuning a UNIX™ domain manager to handle large numbers of fault-tolerant agents.....	463
Tuning job processing on a workstation.....	464
Tuning plan replication.....	465
Tuning the database.....	466
Optimizing the replication of the Symphony file in the database.....	466
Tuning the WebSphere Application Server Liberty Base.....	466
Inadequate Java™ heap size.....	467
Too many manual job submissions.....	467
Too many file dependency checks.....	467
Network configuration availability.....	467
Workload spreading.....	468
Improving job-processing performance.....	468
Mailbox caching - advantages and disadvantages....	468
Setting the synch level parameter.....	469
The fault-tolerant switch manager - impact on performance.....	470
Network Traffic.....	470
Disk Space.....	471
Scalability.....	471
Impact on JnextPlan.....	471
Impact on reporting.....	472
Impact on event rule deployment.....	472
Increasing application server heap size.....	472
Increasing maximum DB2® log capacity.....	473
Oracle tablespace size.....	477
Multiple Dynamic Workload Console production plan reports.....	477
Dynamic Workload Console - adjusting session timeout settings.....	477
Dynamic Workload Console - Increasing application server heap size.....	478
Dynamic Workload Console graphical views.....	480
Chapter 11. Availability.....	481
Resolving user ID account on Windows® operating systems.....	481
Using a temporary directory on UNIX™.....	482
Chapter 12. License Management in IBM License Metric Tool.....	483
Processor Value Unit license model.....	483
Per Job license model.....	487
Using per job queries when upgrading from a version earlier than 9.4 Fix Pack 2.....	493
Notices.....	cdxciv
Index.....

List of Figures

Figure 1: And end-to-end environment configured for high availability..... 122

Figure 2: Realm name in the WebSphere administrative console..... 143

Figure 3: Export of the ltpa keys file..... 143

Figure 4: Realm name in the authentication template..... 144

Figure 5: Password in XOR format..... 145

Figure 6: IBM Workload Scheduler network domain structure..... 264

Figure 7: Symphony file synchronization..... 274

Figure 8: Process creation on domain manager and fault-tolerant agent..... 275

Figure 9: Typical IBM Workload Scheduler network flows..... 280

Figure 10: Overview of keys distribution between master domain manager and dynamic agent..... 304

Figure 11: SSL server and client keys..... 310

Figure 12: Overview of keys distribution between MDM and DWC..... 312

List of Tables

Table 1: Workload service assurance feature.....	17	Table 27: Interaction between <code>enWorkloadServiceAssurance</code> and <code>enWhatIf</code> global options.....	165
Table 2: Condition-based workflow automation.....	18	Table 28: Security object types.....	181
Table 3: Event-driven workload automation feature - general.....	19	Table 29: Actions that users or groups can perform on the different objects.....	183
Table 4: Event-driven workload automation feature - event mailing.....	19	Table 30: Actions that users or groups can perform when designing and monitoring the workload.....	185
Table 5: Event-driven workload automation feature - IBM® Z Workload Scheduler plug-in.....	20	Table 31: Actions that users or groups can perform when modifying current plan.....	185
Table 6: SSL.....	20	Table 32: Actions that users or groups can perform when submitting workload	186
Table 7: Job management.....	21	Table 33: Actions that users or groups can perform when managing the workload environment.....	187
Table 8: Job stream management.....	21	Table 34: Actions that users or groups can perform when managing event rules.....	188
Table 9: Stageman.....	21	Table 35: Administrative tasks that users or groups can perform	189
Table 10: Planman.....	22	Table 36: Actions that users or groups can perform on workload reports.....	189
Table 11: Logging and auditing.....	22	Table 37: Actions that users or groups can perform on Application Lab.....	190
Table 12: Cross dependencies.....	23	Table 38: Actions that users or groups can perform on folders.....	190
Table 13: Open Services for Lifecycle Collaboration (OSLC).....	23	Table 39: Attributes for security object types.....	191
Table 14: SmartCloud Control Desk.....	24	Table 40: Object attribute types for each object type.....	214
Table 15: ServiceNow.....	24	Table 41: Access keywords for composer actions.....	221
Table 16: Automatic failover.....	25	Table 42: Actions - access keywords.....	224
Table 17: Licensing configuration.....	26	Table 43: Calendar - additional access keywords.....	225
Table 18: General.....	28	Table 44: Cpus - additional access keywords.....	225
Table 19: Valid internal job states.....	31	Table 45: Events - access keywords.....	227
Table 20: Valid encryption cipher classes.....	60	Table 46: Files - access keywords.....	228
Table 21: Agent configuration parameters.....	96	Table 47: folders - access keywords.....	229
Table 22: <code>J2EEJobExecutorConfig.properties</code> file keywords.....	107	Table 48: Jobs - additional access keywords.....	231
Table 23: Configuration files for job types with advanced options.....	114		
Table 24: Menu and Group Permissions.....	139		
Table 25: Syntax for special characters.....	154		
Table 26: Variables used in the URL definition.....	154		

Table 49: Parameters - additional access keywords.....	235	Table 78: Elements in UserInfo type.....	378
Table 50: Prompts - additional access keywords.....	236	Table 79: Complete procedure for switching a domain manager in case of a planned outage.....	393
Table 51: Files- access keywords.....	236	Table 80: Complete procedure for switching a domain manager after an unplanned outage.....	395
Table 52: Resources - additional access keywords.....	237	Table 81: Configurable properties for automatic switch broker process.....	407
Table 53: Run cycle groups- access keywords.....	238	Table 82: If and where password changes are required....	413
Table 54: Job streams - additional access keywords.....	238	Table 83: Correspondence between wastools and templates.....	429
Table 55: Users - additional access keywords.....	239	Table 84: Configuration files for job types with advanced options.....	455
Table 56: Variable tables - access keywords.....	240	Table 85: Options for tuning job processing on a workstation.....	464
Table 57: Workload applications - access keywords.....	241	Table 86: Chargeable software components automatically detected by License Metric Tool.....	483
Table 58: Configuration settings.....	270	Table 87: Chargeable software components that require software tag deployment on managed nodes.....	484
Table 59: Critical flow errors.....	280	Table 88: IBM Workload Scheduler chargeable access methods and application plug-ins.....	486
Table 60: Queue sizing conditions.....	282		
Table 61: Example for the ge operator.....	284		
Table 62: Example for the le operator.....	285		
Table 63: Calculation of internal Symphony table.....	292		
Table 64: Changes allowed in IBM Workload Scheduler keystore and truststore.....	310		
Table 65: Files for Local Options.....	322		
Table 66: Type of communication depending on the securitylevel value.....	323		
Table 67: Algorithm for calculating the approximate size of the plan data in the Symphony file.....	344		
Table 68: Algorithm for calculating the approximate size of the database data in the Symphony file.....	344		
Table 69: Example for the ge operator.....	346		
Table 70: Example for the le operator.....	348		
Table 71: Log and trace file maintenance.....	350		
Table 72: Auditable event properties.....	375		
Table 73: Elements in Action type.....	376		
Table 74: Elements in ObjectInfoList type.....	377		
Table 75: Elements in ObjectInfo type.....	377		
Table 76: Elements in Outcome type.....	378		
Table 77: Elements in UserInfoList type.....	378		

About this publication

IBM Workload Scheduler: Administration Guide provides information about the administration of the main components of IBM Workload Scheduler (often called the *engine*).

What is new in this release

Learn what is new in this release.

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

For information about the APARs that this release addresses, see the IBM Workload Scheduler Release Notes at [IBM Workload Scheduler Release Notes](#) and the Dynamic Workload Console Release Notes at [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*.

Chapter 1. Customizing and configuring IBM Workload Scheduler

After installing the product you can customize it to fit your operational requirements. You can also change the customized values at any time. This chapter describes the optional customization steps for IBM Workload Scheduler. It is divided into the following sections:

- [Setting global options on page 14](#)
- [Setting local options on page 51](#)
- [Setting user options on page 76](#)
- [Configuring the dynamic workload broker server on the master domain manager and dynamic domain manager on page 96](#)
- [Configuring the agent on page 77](#)
- [Configuring command-line client access authentication on page 117](#)
- [IBM Workload Scheduler console messages and prompts on page 125](#)

For more information, see the sections about automating production plan processing and managing the production cycle in *User's Guide and Reference*.

Personalizing UI labels

IBM® Workload Scheduler provides the capability to customize user interface labels.

Before you begin

You might find this feature useful for your business users so that the tasks they perform are in the context of your line of business. You can personalize the UI labels for the following UIs:

- Self-Service Catalog and Self-Service Dashboards mobile applications

About this task

The properties file, `whitelabelling.properties`, from which you can modify UI labels must be created manually in a sub-folder named, `Labels`, which you must also create manually in the following path: `<DWC_DATA>usr/servers/registry` directory.

1. Create a new sub-directory named `Labels` in the following path:

On Windows:

```
<DWC_DATA>/usr/servers/dwcserver/registry
```

On UNIX:

```
//<DWC_DATA>/usr/servers/dwcservers/registry
```

2. In the `whitelabelling.properties` file, you can customize your Dynamic Workload Console by adding `dwc.title` = "new title".

3. Add the following parameters to the `whitelabelling.properties` file and assign a value to the labels you want to modify.

```
mobile.title=<value>
ssc.title=<value>
ssd.title=<value>
```

where `<value>` corresponds to the following labels:

Self-Service Catalog and Self-Service Dashboards

Replace `<value>` with the text to replace the current label:

- **mobile.title=** `<value>` If defined, this label will appear instead of "IBM Workload Scheduler Mobile Apps"
- **ssc.title=** `<value>` If defined, this label replaces "Self-Service Catalog"
- **ssd.title=** `<value>` If defined, this label replaces "Self-Service Dashboards"

4. Save your changes.

Setting global options

Manages the IBM Workload Scheduler global options. You can list, show and change them.

Authorization

You must have the following security permissions for the global options file in the IBM Workload Scheduler security file to work with this command:

- For `optman ls` or `optman show`:

```
FILE NAME=GLOBALOPTS ACCESS=DISPLAY
```

- For `optman chg`:

```
FILE NAME=GLOBALOPTS ACCESS=MODIFY
```

See [Configuring user authorization \(Security file\) on page 168](#) for more information on the security file.

Syntax

```
optman [-u | -v]
```

```
optman [connectionParams] chg {option | shortName} = value
```

```
optman [connectionParams] ls
```

```
optman [connectionParams] show {option | shortName}
```

Arguments

connectionParams

If you are using **optman** from the master domain manager, the connection parameters were configured at installation and do not need to be supplied, unless you do not want to use the default values.

If you are using **optman** from the command line client on another workstation, the connection parameters might be supplied by one or more of these methods:

- Stored in the `localopts` file
- Stored in the `useropts` file
- Supplied to the command in a parameter file
- Supplied to the command as part of the command string

For full details of the connection parameters see [Configuring command-line client access authentication on page 117](#).

chg {option | shortName} = value

Change the value of an option to the new value supplied. The option can either be identified by its full or its short name. See [Global options - summary on page 16](#) for a table showing all of the options with their full and short names, value ranges and default values. See [Global options - detailed description on page 29](#) for a full description of each option.

ls

Lists the current values of all global options.

show {option | shortName}

Displays the current value of the indicated option. The option can either be identified by its full or its short name. See [Global options - summary on page 16](#) for a table showing all of the options with their full and short names, value ranges and default values. See [Global options - detailed description on page 29](#) for a full description of each option.

Comments

Some of the changes are effective immediately, but others require a specific action, such as running JnextPlan, restarting the WebSphere Application Server Liberty Base. These actions are indicated in the option descriptions. See *User's Guide and Reference* for more information on the JnextPlan command.

Users can decide to maintain an audit trail recording any changes they perform and the related justifications. To enable the justification option, set up in a system shell the IBM Workload Scheduler environment variables listed below before running any **optman** commands:

IWS_DESCRIPTION

Specify the description to be recorded for each change performed by commands in the shell. The maximum length for this value is 512 characters. A warning message displays if you exceed the maximum and excess characters are truncated.

IWS_CATEGORY

Specify the category to be recorded for each change performed by commands in the shell. The maximum length for this value is 128 characters. A warning message displays if you exceed the maximum and excess characters are truncated.

IWS_TICKET

Specify the ticket to be recorded for each change performed by commands in the shell. The maximum length for this value is 128 characters. A warning message displays if you exceed the maximum and excess characters are truncated.

For more information about the justification option, see the section about keeping track of changes in *Dynamic Workload Console User's Guide*.

Example**Examples****Example 1: list the global options**

To list all of the global options, when your connection parameters are supplied via the `localopts` and `useropts` files, give the following command:

```
optman ls
```

Example 2: show the value of a global option

To show the current value of the `enCarryForward` global option, identifying it by its short name, give the following command:

```
optman show cf
```

Example 3: change the value of a global option

To change the current value of the `enCarryForward` global option, identifying it by its full name, give the following command:

```
optman chg enCarryForward no
```

Global options - summary

This section summarizes the global options that are managed by `optman`. The columns in the tables have the following meanings:

Description

The brief description of the option

Name

The option as used in the `optman` commands.

Short name

The `shortName` as used in the `optman` commands.

Default

The default value that is applied to the option at installation (if present).

Range

The range or choice of values you can supply (where appropriate).

Units

The units that apply to the default and range.

Effect

How to make any changes effective. The following codes have been used:

E

If you are enabling the option, start the Event Processor. If you are disabling the option, stop the Event Processor.

Imm

The change is effective immediately

Imm (DB)

The change is effective immediately in the database only.

J

Run JnextPlan.

J (Plan)

Run JnextPlan - it makes the change effective in the plan only.

NSJ

The change is effective on the next submit job stream action.

NSM

The change is effective on the next send mail action.

NOC

The change is effective on the next change performed on a security object.

W

Restart WebSphere Application Server Liberty Base.

The following tables summarize the global options for managing the features and functions of IBM Workload Scheduler:

Table 1. Workload service assurance feature

Description	Name	Short name	Default	Range	Units	Effect
Enable workload service assurance	enWorkloadServiceAssurance	wa	yes	yes, no	boolean	J
Approaching late offset	approachingLateOffset	al	120	>=0	seconds	J or W
Deadline offset	deadlineOffset	do	2	>=0	minutes	J or W

Table 1. Workload service assurance feature (continued)

Description	Name	Short name	Default	Range	Units	Effect
Promotion offset	promotionOffset	po	120	>=0	seconds	J
Enable forecast start time calculation	enForecastStartTime	st	no	yes, no	boolean	imm

Table 2. Condition-based workflow automation

Description	Name	Short name	Default	Range	Units	Effect
Name of the job which is automatically added to the plan to run the file monitoring task.	fileStartConditionJobName	fc	file_Start Cond	40 bytes		Imm
Name of the job which is automatically added to the plan to resubmit a new instance of the job stream where the start condition is defined.	resubmitJobName	rj	restart_S tartCond	40 bytes		Imm
Default offset set for the start condition deadline.	startConditionDeadlineOffset	cd	2400	0001 - 9959	hhmm	Imm
Prevent job streams from completing in error when the start condition is not met	enStartCondSuccOnDeadline	od	Fr esh inst allat ion	yes - no	boolean	J
			yes			
			Upgr ade			
			no			

Table 3. Event-driven workload automation feature - general

Description	Name	Sh ort n ame	Default	Range	Units	Effect
Enable event driven workload automation	enEventDrivenWorkloadAutomation	ed	yes	yes, no	boolean	J or E
Rules deployment frequency	deploymentFrequency	df	5	0-60	minutes	Imm
Enable event processor HTTPS protocol	enEventProcessorHttpsProtocol	eh	yes	yes, no	boolean	J
IBM event integration facility port for SSL	eventProcessorEIFSslPort	ef	31131	0 - 65535	port number	W and J
IBM event integration facility port	eventProcessorEIFPort	ee	31131	0 - 65535	port number	W and J
EIF Probe server name (used both for events in TEC and TBSM formats)	TECServerName	th	localhost		name	J
EIF Probe server port (used both for events in TEC and TBSM formats)	TECServerPort	tp	5529	0 - 65535	port number	J

Table 4. Event-driven workload automation feature - event mailing

Description	Name	Sh ort n ame	Default	Range	Units	Effect
Mail sender name	mailSenderName	ms	TWS		name	NSM
SMTP server name	smtpServerName	sn	localhost		name	Imm
SMTP Server port	smtpServerPort	sp	25	0 - 65535	port number	NSM
Mail plug-in uses SMTP authentication	smtpUseAuthentication	ua	no	yes, no	boolean	Imm
SMTP user name	smtpUserName	un	<TWS_user>		name	Imm
SMTP user password	smtpUserPassword	up				Imm
Mail plug-in uses SSL	smtpUseSSL	us	no	yes, no	boolean	Imm

Table 4. Event-driven workload automation feature - event mailing (continued)

Description	Name	Short name	Default	Range	Units	Effect
Mail plug-in uses TLS protocol	smtpUseTLS	tl	no	yes, no	boolean	Imm

Table 5. Event-driven workload automation feature - IBM® Z Workload Scheduler plug-in

Description	Name	Short name	Default	Range	Units	Effect
IBM Z Workload Scheduler connector remote server name	zOSRemoteServerName	zr			name	NSJ
IBM Z Workload Scheduler connector server name	zOSServerName	zs	localhost		name	NSJ
IBM Z Workload Scheduler connector server port	zOSServerPort	zp	31217	0 65535	port number	NSJ
IBM® Z Workload Scheduler connector user name	zOSUserName	zu	<TWS_user>		name	NSJ
IBM® Z Workload Scheduler connector user password	zOSUserPassword	zw				NSJ

Table 6. SSL

Description	Name	Short name	Default	Range	Units	Effect
Enable the SSL full connection	enSSLFullConnection	sf	no	yes, no	boolean	J
Enable strong password encryption	enStrEncrypt	se	no	yes, no	boolean	J

Table 7. Job management

Description	Name	Sh ort n ame	Default	Range	Units	Effect
Maximum prompts after abend	baseRecPrompt	bp	1000	0 65535	prompts	J
Additional prompts after abend	extRecPrompt	xp	1000	0 65535	prompts	J
Concurrent access to resources	enExpandedResources	er	yes	yes, no	boolean	J
Automatically grant logon as batch	enLogonBatch	lb	no	yes, no	boolean	J
Long duration job threshold	longDurationThreshold	ld	150	100 - 1000	seconds	J or W
User for binding to remote jobs from shadow job	bindUser	bu	<TWS_us er>			Imm

Table 8. Job stream management

Description	Name	Sh ort n ame	Default	Range	Units	Effect
Job streams without jobs policy	enEmptySchedsAreSucc	es	no	yes, no	boolean	J
Prevent job stream without "at" dependency from starting	enPreventStart	ps	yes	yes, no	boolean	J

Table 9. Stageman

Description	Name	Sh ort n ame	Default	Range	Units	Effect
Carry job states	carryStates	cs	null		list of states	J
Enable carry forward	enCarryForward	cf	all	all, no	boolean	J
Enable carry forward for internetwork dependencies	enCFinterNetworkDeps	ci	yes	yes, no	boolean	J

Table 9. Stageman (continued)

Description	Name	Short name	Default	Range	Units	Effect
Enable carry forward resource quantity	enCFResourceQuantity	rq	yes	yes, no	boolean	J
Retain rerun job name	enRetainNameOnRerunFrom	rr	no	yes, no	boolean	J
Remove obsolete job streams	untilDays	ud	0	>=0	days	J

Table 10. Planman

Description	Name	Short name	Default	Range	Units	Effect
Maximum preproduction plan length	maxLen	xl	8	8 - 365	days	J
Minimum preproduction plan length	minLen	ml	8	7 - 365	days	J

Table 11. Logging and auditing

Description	Name	Short name	Default	Range	Units	Effect
Log cleanup frequency	logCleanupFrequency	lc	5	0 - 60	minutes	J
Log history period	logHistory	lh	10	>=0	days	J
Logman minimum and maximum run time policy	logmanMinMaxPolicy	lm	both		literal	J
Logman normal run time calculation policy	logmanSmoothPolicy	lt	-1	0 - 100	factor	J
Enable database auditing	enDbAudit	da	0	0, 1	boolean	Imm
Type of store to be used to log database audit records	auditStore	as	file	db, file, both		Imm
Audit history period	auditHistory	ah	180	>=1	days	Imm

Table 11. Logging and auditing (continued)

Description	Name	Short name	Default	Range	Units	Effect
Enable auditing of database GET operations.	enDbGetOpsAudit	dg	1	0, 1	boolean	Imm

Table 12. Cross dependencies

Description	Name	Short name	Default	Range	Units	Effect
Number of days for retrying to send notifications about job status changes to the remote engine if the notification fails	notificationTimeout	nt	5	1-90	Number	Imm

Table 13. Open Services for Lifecycle Collaboration (OSLC)

Description	Name	Short name	Default	Range	Units	Effect
Description of the IBM Workload Scheduler automation service provider	oslcAutomationDescription	ad			name	Imm
Title of the IBM Workload Scheduler automation service provider	oslcAutomationTitle	at			name	Imm
Host name of the IBM Workload Scheduler service provider (host name of the active master domain manager)	oslcProviderUri	pu			name	Imm
Description of the IBM Workload Scheduler provisioning service provider	oslcProvisioningDescription	pd			name	Imm
Title of the IBM Workload Scheduler provisioning service provider	oslcProvisioningTitle	pt			name	Imm

Table 13. Open Services for Lifecycle Collaboration (OSLC) (continued)

Description	Name	Sh ort n ame	Default	Range	Units	Effect
Password associated with the user who connects to the Registry Services	oslcRegistryPassword	rp			name	Imm
Address of the Registry Services	oslcRegistryUri	cu			name	Imm
User who connects to the Registry Services	oslcRegistryUser	ru			name	Imm

Table 14. SmartCloud Control Desk

Description	Name	Sh ort n ame	Default	Range	Units	Effect
Address of the SmartCloud Control Desk	sccdUrl	du			name	Imm
User who connects to the SmartCloud Control Desk	sccdUserName	dn			name	Imm
Password associated with the user who connects to the SmartCloud Control Desk	sccdUserPassword	dp			name	Imm

Table 15. ServiceNow

Description	Name	Sh ort n ame	Default	Range	Units	Effect
Address of the ServiceNow server	servicenowUrl	nu			name	Imm
User who connects to the ServiceNow server	servicenowUserName	nn			name	Imm
Password associated with the user who connects to the ServiceNow server	servicenowUserPassword	np			name	Imm

Table 16. Automatic failover

Description	Name	Short name	Default	Range	Units	Effect
Enables or disables the automatic failover feature which invokes an automatic switch from the master domain manager, event manager, or both, to a backup.	enAutomaticFailover	af	yes	yes, no	boolean	W
Enables or disables automatic failover actions, such as, the automatic switch of the master or automatic restart of the fault-tolerant agent. This option takes effect only if the enAutomaticFailover option is set to <i>yes</i> .	enAutomaticFailoverActions	aa	yes	yes, no	boolean	W
A comma-separated list of workstation names, including the current event manager workstation, that serve as backups for the event manager workstation when the automatic failover feature is enabled.	workstationEventMgrListInAutomaticFailover	we		comma-separated list of workstation names. The maximum length is 256 bytes.	name	W
A comma-separated list of workstations, including the current master domain manager, that serve as backups for the master domain manager when the automatic failover feature is enabled.	workstationMasterListInAutomaticFailover	wm		comma-separated list of workstation names. The maximum	name	W

Table 16. Automatic failover (continued)

Description	Name	Short name	Default	Range	Units	Effect
				minimum length is 256 bytes.		

Table 17. Licensing configuration

Description	Name	Short name	Default	Range	Units	Effect
Type of accepted license for IBM Workload Scheduler	licenseType	ln	ws	ws, workstation		J
Specify the default license type for IBM Workload Scheduler workstations.	defaultWksLicenseType	wn	Infrastructure environment processor job objects binaries resources	• P E R S E R VER • P E R JOB		J

Table 17. Licensing configuration (continued)

Description	Name	Short name	Default	Range	Units	Effect
			t a l l a t ion and p e r S e r v e rin u p g r a de.			
			In an on- premi ses enviro nm ent			
			p e r S e			

Table 17. Licensing configuration (continued)

Description	Name	Short name	Default	Range	Units	Effect
			10			
			10			

Table 18. General

Description	Name	Short name	Default	Range	Units	Effect
Company name	companyName	cn			name	J
Delete folders	folderDays	fd	10	0 - 10	days	J
Enable centralized security in the classic security model	enCentSec	ts	no	yes, no	boolean	J
Evaluate start-of-day	enLegacyStartOfDayEvaluation	le	no	yes, no	boolean	J
Enable list security check	enListSecChk	sc	no	yes, no	boolean	J (Plan) Imm (DB)
Enable plan auditing	enPlanAudit	pa	0	0, 1	boolean	J
Enable security file creation in the role-based security model	enRoleBasedSecurityFileCreation	rs	no	yes, no	boolean	Imm
Enable extended field support in the security file.	enSecFileExtendedFields	sl	no	yes, no	boolean	NOC
Enable the fault-tolerant switch manager	enSwfaultTol	sw	no	yes, no	boolean	J
Enable time zones	enTimeZone (deprecated)	tz	yes	yes, no	boolean	J (Plan) Imm (DB)
Enable What-if Analysis	enWhatIfAnalysis	wi	yes	yes, no	boolean	J
Ignore calendars	ignoreCals	ic	no	yes, no	boolean	J
Start time of processing day	startOfDay	sd	0000	0000 2359	hhmm	J
Job statistics history period	statsHistory	sh	10	>=0	days	J (Plan)

Table 18. General (continued)

Description	Name	Sh ort n ame	Default	Range	Units	Effect
						Imm (DB)
Critical Jobs Risk Confidence	riskConfidence	rc	80% in fresh installati on, 50% in upgrade	1-99	Number	Imm

Global options - detailed description

This section gives full descriptions of the global options managed by optman:

approachingLateOffset | al

Approaching late offset. Used in workload service assurance. The critical start time of a job in the critical network is the latest time that the job can start without causing the critical job to finish after the deadline. In most cases, a job will start well before the critical start time so that if the job runs longer than its estimated duration, the situation does not immediately become critical. Therefore, if a job has not started and the critical start time is only a few minutes away, the timely completion of the critical job is considered to be potentially at risk.

The *approachingLateOffset* option allows you to determine the length of time before the critical start time of a job in the critical network at which you are to alerted to this potential risk. If a job has still not started the specified number of seconds before the critical start time, the job is added to a hot list that can be viewed on the Dynamic Workload Console.



Note: To qualify for addition to the hot list, all time and follow dependencies must have been resolved.

This option is only active if *enWorkloadServiceAssurance* is set to *yes*.

The default is 120 seconds.



Note: Whatever value you set for this option, if IBM Workload Scheduler loses the connection with its database, the default value is applied to critical job processing, and the warning message AWSJCO135W is issued to tell you what has happened.

Run JnextPlan or restart WebSphere Application Server Liberty Base (stopappserver and startappserver) to make this change effective.

auditHistory | ah

Audit history period. Used in audit management. This setting applies only when the **auditStore** option is set to **db**. Enter the number of days for which you want to save audit record data. Audit records are discarded on a FIFO (first-in first-out) basis.

The default value is *180* days. This option takes effect immediately.

For more information about auditing, see [Auditing facilities on page 364](#).

auditStore | as

Type of store to be used to log database and plan audit records. Enter one of the following:

file

To specify that a flat file in the `TWA_home/TWS/audit/database` directory is used to store the audit records. This is the default value.

db

To specify that the IBM Workload Scheduler database itself is used to store the audit records.

both

To have audit records logged in both the file and the database.

The default value is `both`. Any change of this value is effective immediately.



Note: When you upgrade the master domain manager from a previous release, the default value for this global option is changed. The default value is now **both**. The reason for this change is to support the auditing feature which introduces reporting, versioning and rollback functions for database objects. If you customized the default value in the previous release, the value is overwritten with the new value with the exception of the **auditStore** option with the **DB** value assigned. If the **auditStore** option was set to **DB**, this value is maintained and is not overwritten.

For more information about auditing, see [Auditing facilities on page 364](#).

baseRecPrompt | bp

Maximum prompts after abend. Specify the maximum number of prompts that can be displayed to the operator after a job abends.

The default value is *1000*. Run JnextPlan to make this change effective.

bindUser | bu

User for binding to remote jobs from shadow job. Specify the user ID that is used to bind a shadow job to a remote job during the security check for "cross dependencies". This user must be given at least the following authorizations in the security file:

- *Display* access to the *job* and *schedule* objects that need to be bound
- *List* access to *job* objects that need to be bound

However, the ID does not need to be in the user registry of the engine, nor have a password, as it is only required for authorization purposes.

The default value is the <TWS_user>. Any change of this value is effective immediately.

carryStates | cs

Carry job states. A reproduction option that affects the operation of the *stageman* command. Specify the jobs, by state, to be included in job streams that are carried forward. Enclose the job states in parentheses, double quotation marks, or single quotation marks. Commas can be replaced by spaces. The valid internal job states are as follows:

Table 19. Valid internal job states

<i>abend</i>	<i>abenp</i>	<i>add</i>	<i>bound</i>	<i>done</i>	<i>error</i>	<i>exec</i>
<i>fail</i>	<i>hold</i>	<i>intro</i>	<i>pend</i>	<i>ready</i>	<i>rjob</i>	<i>sched</i>
<i>skel</i>	<i>succ</i>	<i>succp</i>	<i>suppr</i>	<i>susp</i>	<i>wait</i>	<i>waitd</i>

Some examples of the option are as follows:

```
carryStates="abend,exec,hold,intro"
carryStates='abend,exec,hold,intro'
carryStates="abend, exec, hold, intro"
carryStates='abend, exec, hold, intro'
```

An empty list is entered as follows:

```
carryStates=null
```

The default value is *null*, which corresponds to selecting all states. Run JnextPlan to make this change effective.

companyName | cn

Company name. Specify the name of your company. The maximum length is 40 bytes. If the name contains spaces, enclose the name in double quotation marks ("). If you use the Japanese-Katakana language set, enclose the name within single or double quotation marks.

Run JnextPlan to make this change effective.

deadlineOffset | do

Deadline offset. Used in workload service assurance. Used to calculate the critical start of a critical job in the case where a deadline has not been specified neither for the job nor its job stream. In this case the deadline is defaulted to the plan end date and time, plus this offset, expressed in minutes.

This option is only active if *enWorkloadServiceAssurance* is set to *yes*.

The default is 2 minutes.



Note:



1. **Important:** When the plan is extended, the start time of critical jobs with a deadline calculated with this mechanism is automatically changed as a consequence of the fact that it must now match the new plan finishing time.
2. Whatever value you set for this option, if IBM Workload Scheduler loses the connection with its database, the default value is applied to critical job processing, and the warning message AWSJCO135W is issued to tell you what has happened.

Run JnextPlan or restart WebSphere Application Server Liberty Base (stopappserver and startappserver) to make this change effective.

defaultWksLicenseType / wn

Specify the default licensing model for IBM Workload Scheduler workstations. This option is supported only if the licenseType option is set to `byWorkstation` and a specific value was not specified at creation time for each workstation. For more information, see [licenseType | In on page 41](#). Supported values are as follows:

perServer

to specify the IBM Workload Scheduler Processor Value Unit (PVU) pricing.

perJob

to specify the IBM Workload Scheduler Per Job (PJ) pricing.

The default value varies, depending on the environment type:

In a Docker environment

By default, the defaultWksLicenseType option is set to `perJob` in a fresh installation and to `perServer` during product upgrade. By default, the licenseType option is set to `byWorkstation` in a fresh installation. This setting is not modified during product upgrade. As a result, the value of the defaultWksLicenseType option is applied to the creation of all workstation types.

In an on-premises environment

By default, the licenseType option is set to `perServer`, and also defaultWksLicenseType option is set to `perServer`. Before changing this value, contact your sales representative.

deploymentFrequency | df

Rules deployment frequency. Used in event rule management. Specify the frequency, in minutes, with which rules are to be checked to detect if there are changes to deploy. All active rules (active rules have the `isDraft` property set to `no` in their definition) that have been changed or added since the last deployment are deployed.

Valid values are in the 0-60 minutes range. If you specify 0, the changes are not deployed automatically and you must use the planman deploy command.

The default value is 5 minutes. The change is effective immediately.

enAddUser | au

Enable the automatic user addition into the Symphony file. This option enables the automatic addition of a user into the Symphony file after you create or modify the user in the database. If you specify "yes", the user is automatically added to the Plan. If you specify "no", the user is not automatically added to the Plan.

The default value is "yes". Changes to this parameter are effective immediately.

For more information about how to use this feature, see "IBM Workload Scheduler: User's Guide and Reference".

enAddWorkstation | aw

Enable the automatic dynamic agent, pool, and dynamic pool workstation addition into the Symphony file.

This option enables the automatic addition of a dynamic agent, pool, or dynamic pool workstation into the Symphony file after you created the workstation in the database. If you specify "yes", the workstation is automatically added to the Plan. If you specify "no", the workstation is not automatically added to the Plan.

The default is "no". Changes to this parameter are effective immediately.

For more information about how to use this feature, see *IBM Workload Scheduler: User's Guide and Reference*.

enAutomaticFailover | af

Enable or disable the automatic failover feature. This option enables or disables the automatic failover feature which invokes an automatic switch from the master domain manager, event manager, or both, to a backup workstation. Eligible backups for both the master domain manager and the event manager can be specified using the optman options, workstationMasterListInAutomaticFailover and workstationEventManagerListInAutomaticFailover, respectively.

A fresh installation of IBM® Workload Scheduler V9.5FP2 or later enables this feature by default (*yes*). This feature is disabled (*no*) when you upgrade to IBM® Workload Scheduler V9.5FP2 or later. Changes to this parameter require restarting WebSphere Application Server Liberty Base.

enAutomaticFailoverActions | aa

Enable or disable the automatic failover actions. This option enables or disables automatic failover actions, such as, the automatic switch of the master or the automatic restart of the fault-tolerant agent. This option takes effect only if the enAutomaticFailover option is set to *yes*. You can set this option to *no* in the case of a planned maintenance window.

By default, this option is set to *yes*. Changes to this parameter require restarting WebSphere Application Server Liberty Base.

workstationMasterListInAutomaticFailover | wm

A list of workstations eligible to serve as a backup for the master. A comma-separated list of workstations that serve as backups for the master domain manager, including the current master domain manager itself, when the automatic failover feature is enabled. The maximum length is 256 bytes.

If no workstations are specified in this list, then all backup master domain managers in the domain are considered eligible backups. Changes to this parameter require restarting WebSphere Application Server Liberty Base.

workstationEventMgrListInAutomaticFailover | we

A list of workstations eligible to serve as a backup for the event manager. A comma-separated list of workstations that serve as backups for the event manager, including the current event manager itself, when the automatic failover feature is enabled. The maximum length is 256 bytes.

If no workstations are specified in this list, then all backup master domain managers in the domain are eligible backups. Changes to this parameter require restarting WebSphere Application Server Liberty Base.

enCarryForward | cf

Enable carry forward. A preproduction option that affects the operation of the *stageman* command. Specify if job streams that did not complete are carried forward from the old to the new production plan (Symphony). Enter *yes* to have incompleting job streams carried forward only if the *Carry Forward* option is enabled in the Job Scheduler definition. Enter *all* to have all incompleting job streams carried forward, regardless of the *Carry Forward* option. Enter *no* to completely disable the *Carry Forward* function. If you run the `JnextPlan -for 0000` command and the *Carry Forward* option is set to either *yes* or *no*, a message is displayed informing you that incompleting job streams might not be carried forward. When the *stageman -carryforward* command is used, it overrides *enCarryForward*. See *IBM Workload Scheduler: User's Guide and Reference* for more information. If this option is set to *no*, running jobs are moved to the USERJOBS job stream.

The default value is *all*. Run *JnextPlan* to make this change effective.

enCentSec | ts

Enable centralized security. In the classic security model, determine, how the security file is used within the network. Centralized security is not relevant to an end-to-end scheduling environment.

If set to *yes*, the security files of all the workstations of the network can be created and modified only on the master domain manager. In this case, the IBM Workload Scheduler administrator is responsible for their production, maintenance, and distribution.

If set to *no*, the security file of each workstation can be managed by the root user or administrator of the system. The local user can run the *makesec* command to create or update the file.

See *IBM Workload Scheduler: User's Guide and Reference* for more information about centralized security.

The default value is *no*. Run *JnextPlan* to make this change effective.



Note: This option does not apply to role-based security model.

enCFinterNetworkDeps | ci

Enable carry forward for internetwork dependencies. A preproduction option that affects the way *stageman* handles internetwork dependencies. It specifies if external job streams are carried forward from the old to the

new production plan (Symphony file). Enter *yes* to have all external job streams carried forward. Enter *no* to have no external job streams carried forward.

The default value is *yes*. Run JnextPlan to make this change effective.

enCFResourceQuantity | rq

Enable carry forward resource quantity. A preproduction option that affects the way stageman handles resources. Enter *yes* to carry forward the resource quantity from the old production file to the new. Enter *no* to not carry forward the resource quantity. Stageman carries forward resource quantities only if the resource is needed by a job or job stream that is also being carried forward. Otherwise the resource quantities are set to the original value. See *IBM Workload Scheduler: User's Guide and Reference* for details on using this feature.

The default value is *yes*. Run JnextPlan to make this change effective.

enDbAudit | da

Enable auditing on information available in the database. Enable or disable auditing on information available in the database. To enable auditing on information available in the database, specify *1*. To disable auditing on information available in the database, specify *0*. Auditing information is logged to a flat file in the `TWA_home/TWS/audit/database` directory, to the IBM Workload Scheduler database itself, or to both. To choose which, set the optman property `auditStore`. Each IBM Workload Scheduler workstation maintains its own log. Only actions are logged, not the success or failure of the action. Installation of dynamic domain managers and agents is not recorded in audit logs.

The default value is *1*. Changes to this parameter are effective immediately.



Note: When you upgrade the master domain manager from a previous release, the default value for this global option is changed. The default value is now **1**. The reason for this change is to support the auditing feature which introduces reporting, versioning and rollback functions for database objects. If you customized the default value in the previous release, the value is overwritten with the new value.

For more information about auditing, see [Auditing facilities on page 364](#).

enDbGetOpsAudit

Enable auditing of GET database operations Enable or disable auditing on database GET operations. Disabling this auditing feature might improve performance. All other database operations are not affected. To disable auditing on database GET operations, specify *0*. To enable auditing on database GET operations, specify *1*. This parameter is effective only if general database audit is enabled (**enDbAudit=1**).

The default value is *1*. Changes to this parameter are effective immediately.

enEmptySchedsAreSucc | es

Job streams without jobs policy. Specify the behavior of job streams without any jobs. If set to *yes*, the job streams that contain no jobs are set to SUCC after their dependencies are resolved. If set to *no*, the job streams are left in READY status.

The default value is *no*. Run JnextPlan to make this change effective.

enEventDrivenWorkloadAutomation | ed

Enable event-driven workload automation. Enable or disable the event-driven workload automation feature. To enable, specify *yes*. To disable, specify *no*.

The default value is *yes*.

After disabling, you must run JnextPlan and stop the event processing server (with the conman `stopevtp` command).

After enabling, you must run JnextPlan and start the event processing server (with the conman `startevtp` command).

enEventDrivenWorkloadAutomationProxy | pr

Enable event-driven workload automation proxy. Enable or disable the event-driven workload automation proxy feature. To enable, specify *yes*. To disable, specify *no*.

The default value is *no*. Run JnextPlan to make this change effective.

enEventProcessorHttpsProtocol | eh

Enable event processor HTTPS protocol. Used in event rule management. Enables or disables the use of the HTTPS protocol to connect to the event processor server. To enable, enter *yes*. To disable, enter *no*.

The default value is *yes*. Run JnextPlan to make this change effective.

enExpandedResources

Enables up to 60 concurrent holders for an IBM Workload Scheduler resource. Enter *yes* to enable up to 60 concurrent holders for a resource. Enter *no* to disable the feature and use only 32 holders for a resource.

The default value is *yes*. Run JnextPlan to make this change effective.

enForecastStartTime | st

Enable forecast start time. Only applicable when workload service assurance is enabled (see *enWorkloadServiceAssurance*). Enter *yes* to enable the calculation of the predicted start time of each job when running a forecast plan: this option is recommended if you want to take advantage of the enhanced forecast capability that calculates the start time of each job considering the estimated duration of its predecessor jobs. Enabling this feature could negatively impact the time taken to generate the forecast plan. Enter *no* to disable the calculation of the predicted start time of each job when running a forecast plan.

The default value is *no*. Any change of this value is effective immediately.

When this option is set to *yes*, the **enPreventStart** global option is ignored during the creation of forecast plans.

enLegacyStartOfDayEvaluation | le

Evaluate start-of-day. Specify how the *startOfDay* option is to be managed across the IBM Workload Scheduler network. This is a legacy setting and should always be set to *no* starting from release 9.4.0 and later. If you set

this option to *yes*, the *startOfDay* value on the master domain manager is converted to the local time zone set on each workstation across the network. If you set this option to *no*, the *startOfDay* value on the master domain manager is applied as is on each workstation across the network. This option requires that the *enTimeZone* option is set to *yes* to become operational.

The default value is *no*. Run JnextPlan to make this change effective.

enListSecChk | sc

Enable list security check. Control the objects in the database and the plan that a user is permitted to list when running a query on the Dynamic Workload Console or IBM® Workload Scheduler database, for example running a composer list, or a conman show command. If set to *yes*, objects in the plan returned from a query or show command are shown to the user only if the user has been granted the list permission in the security file. If set to *no*, all objects are shown, regardless of the settings in the security file.



Note: Setting this option to *yes* affects how the graphical user interfaces function for the users defined in the security file.

The default value is *no*. Run JnextPlan to make this change effective for the plan. For the database, this option takes immediate effect.

enLogonBatch | lb

Automatically grant logon as batch. This is for Windows® jobs only. If set to *yes*, the logon users for Windows® jobs are automatically granted the right to *Logon as batch job*. If set to *no*, or omitted, the right must be granted manually to each user or group. The right cannot be granted automatically for users running jobs on a backup domain manager, so you must grant those rights manually.

The default value is *no*. Run JnextPlan to make this change effective.

enPlanAudit | pa

Enable plan auditing. Enable or disable auditing on information available in the plan. To enable auditing on information available in the plan, specify *1*. To disable auditing on information available in the plan, specify *0*. Auditing information is logged to a flat file, to the IBM Workload Scheduler database itself, or to both. To define the logging location, set the **auditStore** global option. For more information, see [auditStore | as on page 30](#).

The audit file is located in the following path:

```
TWA_home\TWS\audit\database
```

```
TWA_DATA_DIR/audit/database
```

Each IBM Workload Scheduler workstation maintains its own log. For the plan, only actions are logged in the auditing file, not the success or failure of any action.

For more information about auditing, see [Auditing facilities on page 364](#).

The default value is *1*. Changes to this parameter are effective immediately.



Note: When you upgrade the master domain manager from a previous release, the default value for this global option is changed. The default value is now **1**. The reason for this change is to support the auditing feature which introduces reporting, versioning and rollback functions for database objects. If you customized the default value in the previous release, the value is overwritten with the new value.

enPreventStart | ps

Prevent job stream without "at" dependency from starting. Specify if job streams without an *at* dependency are to be prevented from starting immediately, without waiting for the beginning of the day the run cycle specified in the Job Scheduler identifies. Valid values are *yes* and *no*.

The default value is *yes*. Run JnextPlan to make this change effective.

When the **enForecastStartTime** option is set to *yes*, this option is ignored during the creation of forecast plans.

enRetainNameOnRerunFrom | rr

Retain rerun job name. A production option that affects the operation of Batchman, the production control process of IBM Workload Scheduler. Its setting determines if jobs that are rerun with the Conman *rerun* command retain their original job names. To have rerun jobs retain their original job names, enter *yes*. Enter *no* to assign the *rerun from* name to rerun jobs.

The default value is *no*. Run JnextPlan to make this change effective.

enRoleBasedSecurityFileCreation | rs

Enable the role-based security model. This option enables the automatic creation of the security file using the role-based security model. You define the role-based security model in the master domain manager database by using the **Manage Workload Security** interface from Dynamic Workload Console or the **composer** command-line program.

The default value is *no*, which means that the role-based security model is not enabled for your installation. You continue to use the classic security model that allows you to update your security file by using *dumpsec* and *makesec* commands from the command line.

At any time, specify *yes* if you want to enable the role-based security model and replace your current security file. A new security file is created and updated with the security objects (domains, roles, and access control lists) that you define in the master domain manager database by using the **Manage Workload Security** interface from Dynamic Workload Console.

For more information about how to use this option, see [Configuring user authorization \(Security file\) on page 168](#).

Changes to this parameter are effective immediately.

enSecFileExtendedFields

Enable extended fields support in the security file. Enable long attribute values for all scheduling objects in the security file. When this option is enabled, it permits the use of the extended version of the security file with the attribute field value length set to 64K rather than 255 bytes.

The default value is *no*. This change becomes effective the first time you edit a security object.

enSSLFullConnection | sf

Enable the SSL full connection. Specify that IBM Workload Scheduler uses a higher level of SSL connection than the standard level. For full details see [Configuring full SSL security on page 324](#). Valid values are *yes* to enable the SSL full connection or *no* to disable the SSL full connection.

The default value is *no*. Run JnextPlan to make this change effective.

enStartCondSuccOnDeadline | od

Prevent job streams from completing in error when the start condition is not met Specify the behavior of the job stream when the start condition is not met and the **Start once** option is not selected. If you set the **Start once** option, this option is ignored. If you set the option to *yes*, when the deadline for the start condition is met, the monitoring job is confirmed in **Successful** status and the job stream is canceled. If you set the option to *no*, the monitoring job is killed, so both the monitoring job and the job stream change to **Error** status.

The default value is *yes* in a fresh installation and *no* in upgrade to maintain compatibility with previous versions. Run JnextPlan to make this change effective.

enStrEncrypt | se

Enable strong password encryption. Enable or disable strong encryption. Enable strong encryption by setting this option to *yes*. See [Configuring the SSL connection protocol for the network on page 323](#).

The default value is *no*. Run JnextPlan to make this change effective.

enSwfaultTol | sw

Enable the fault-tolerant switch manager. Enable or disable the fault-tolerant switch manager feature. Valid values are *yes* to enable the fault tolerant switch manager, and *no* to disable it. This option has not dynamic capabilities and is not designed to work with broker agents. It applies to fault-tolerant agents. See the *IBM Workload Scheduler: User's Guide and Reference* for more details.

The default value is *no*. Run JnextPlan to make this change effective.

enTimeZone | tz

Enable time zones. Enables the time zone option.

enWhatIfAnalysis | wi

Enable What-if Analysis. Enables or disables What-if Analysis, which is the feature that shows plan activities displayed against time and give you a visual representation of your plan at a glance in real time. To enable What-if Analysis, specify *yes*. To disable What-if Analysis, specify *no*. See *Dynamic Workload Console User's Guide* for details on using this feature..

The default value is *yes*. Run JnextPlan to make this change effective.

enWorkloadServiceAssurance | wa

Enable workload service assurance. Enables or disables workload service assurance, which is the feature that manages the privileged processing of mission critical jobs and their predecessors. Specify *yes* to enable or *no* to disable.



Note: Before starting to use workload service assurance you must set up the `<TWS_user>` in the security file to have the appropriate access to the objects that this feature will modify - see [The <TWS_user> - special security file considerations on page 241](#)

The default value is *yes*. Run JnextPlan to make this change effective.

eventProcessorEIFSslPort | ef

Tivoli® event integration facility port. Used in event rule management. Specify the port number for SSL where the event processor server receives events from the Tivoli® Event Integration Facility (EIF). Valid values are in the *0-65535* range.

The default value is *31131*. If you change the value, restart WebSphere Application Server Liberty Base (stopappserver and startappserver) and run JnextPlan to make this change effective.

eventProcessorEIFPort | ee

Tivoli® event integration facility port. Used in event rule management. Specify the port number where the event processor server receives events from the Tivoli® Event Integration Facility (EIF). Valid values are in the *0-65535* range.

The default value is *31131*. If you change the value, restart WebSphere Application Server Liberty Base (stopappserver and startappserver) and run JnextPlan to make this change effective.

If you use a security firewall, make sure this port is open for incoming and outgoing connections.

extRecPrompt | xp

Additional prompts after abend. Specify an additional number of prompts for the value defined in *baseRecPropmt*. This applies when a job is rerun after abending and the limit specified in *baseRecPropmt* has been reached.

The default value is *1000*. Run JnextPlan to make this change effective.

fileStartConditionJobName | fc

Name of the job in charge of running the file monitoring task . Applicable only if you select file as the start condition type. Specify the name of the job which is automatically added to the plan to run the file monitoring task. This value is used by default if you do not specify any value for the job name when defining the start condition. If you specify a value for the job name, this value is ignored.

The default value is *FILE_STARTCOND*. The maximum supported length is 40 bytes. Changes to this parameter are effective immediately.

folderDays | fd

Remove deleted folders, prompts, resources, and workstations from the database. When deleting a folder, a prompt, or resource, if there are still objects in the plan that reference these objects, then another folder, prompt, or resource cannot be renamed with the name of the deleted folder, prompt or resource for the number of days specified by "folderDays?". However, a brand new folder, prompt, or resource can be created with the name of the deleted object.

When deleting a workstation, if the workstation is still in the plan, then another workstation cannot be renamed with the name of the deleted workstation for the number of days specified by the global option folderDays. However, a brand new workstation can be created with the name of the deleted workstation. This behavior applies only to dynamic agents, pools, and dynamic pools.

The default value is 10 days.

ignoreCals | ic

Ignore calendars. A reproduction option that affects the operation of the planman command. Its setting determines if user calendars are copied into the new production plan (Symphony) file. To prevent user calendars from being copied into the new production plan, enter yes.

The default value is *no*. See *IBM Workload Scheduler: User's Guide and Reference*. Run JnextPlan to make this change effective.

licenseType | ln

Type of accepted license for IBM Workload Scheduler.

Supported values are:

ws

perServer

to specify the IBM Workload Scheduler Processor Value Unit (PVU) pricing.

wa

perJob

to specify the IBM Workload Scheduler Per Job (PJ) pricing.

byWorkstation

to specify that the licensing type (either **perServer** or **perJob**) is specified at creation time for each workstation. When you specify this value, you can define the default type of license for each workstation by setting the **defaultWksLicenseType** option. For more information, see [defaultWksLicenseType / wn on page 32](#). For more information about defining workstations, see the section about workstation definition in *User's Guide and Reference*.

The default value is **perServer**. Run JnextPlan to make this change effective. For additional information about license management and metrics, see [License Management in IBM License Metric Tool on page 483](#).

You can define this option for the following workstation types:

- master domain manager
- fault-tolerant agent
- standard agent
- dynamic agent

logCleanupFrequency | lc

Log cleanup frequency. Used in event rule and audit management . Specify how often the automatic cleanup of log instances is run. Valid values are in the 0-60 minutes range. If you specify 0, the automatic cleanup feature is disabled.

The default value is 5 minutes. This option takes effect immediately.

logHistory | lh

Log history period. Used in event rule management. Enter the number of days for which you want to save rule instance, action run, and message log data. Log instances are discarded on a FIFO (first-in first-out) basis.

The default value is 10 days. This option takes effect immediately.

logmanMinMaxPolicy | lm

Logman minimum and maximum run times policy. Specify how the minimum and maximum job run times are logged and reported by logman. Possible values are:

elapsedtime

The minimum and maximum elapsed runtimes are logged and reported.

cputime

The minimum and maximum CPU run times are logged and reported.

both

Both the minimum and maximum job runtimes are logged and reported.

See *IBM Workload Scheduler: User's Guide and Reference* for details on using this feature.

The default value is *both*. Run JnextPlan to make this change effective.

logmanSmoothPolicy | lt

Logman normal run time calculation policy. Set the weighting factor that favors the most recent job run when calculating the normal (average) run time for a job. This is expressed as a percentage. For example, specify 40 to apply a weighting factor of 40% to the most recent job run, and 60% to the existing average. See *IBM Workload Scheduler: User's Guide and Reference* for more information about how to use this option.

The default value is 10. Run JnextPlan to make this change effective.

longDurationThreshold | ld

Long duration job threshold. Specify, when comparing the actual duration of a job to the estimated duration, the threshold over which the job is considered to be of "long duration." The threshold value is expressed as a percentage with respect to the estimated duration. For example, if the threshold is set to 150, and the actual

duration is more than 150% of the estimated duration (it is 50% greater), the job is considered to be a "long duration" job.

If you have the workload service assurance feature enabled, the effect of a "critical" job satisfying the long duration criteria is that the job is inserted automatically into the hot list.

Valid values are between:

100

The minimum value. All jobs that exceed the estimated duration are considered long duration jobs

1000

The maximum value. Only those jobs that last ten times as long as their estimated duration are considered as long duration jobs

The default is *150*.



Note: Whatever value you set for this option, if you have the workload service assurance feature enabled, and IBM Workload Scheduler loses the connection with its database, the default value is applied to critical job processing, and the warning message AWSJCO135W is issued to tell you what has happened.

Run JnextPlan or restart WebSphere Application Server Liberty Base (stopappserver and startappserver) to make this change effective.

mailSenderName | ms

Mail sender name. Used in event rule management. If you deploy rules implementing an action that sends emails via an SMTP server, specify a string to be used as the sender of the emails.

The default value is *TWS*. Changes to this parameter are effective for the next mail send action performed.

maxLen | xl

Maximum preproduction plan length. Specify the maximum length of the preproduction plan in days after it is automatically extended or created. The value for *maxLen* must be greater than or equal to the value for *minLen* and must be in the range of 8 to 365.

The default is *14* days. Run JnextPlan to make this change effective.

minLen | ml

Minimum preproduction plan length. Specify the minimum length in days of the preproduction plan that can pass after the production plan is created or extended, without extending the preproduction plan. If the days left in the preproduction plan after a JnextPlan are less than the value of this option, the preproduction plan is automatically extended. The value for *minLen* must be less than or equal to the value for *maxLen* and must be in the range of 7 to 365.

The default is *8* days. Run JnextPlan to make this change effective.

notificationTimeout | nt

Notification timeout. Used in cross dependencies. Specify how many days IBM Workload Scheduler must retry sending notifications about job status changes to the remote engine if the notification fails. When this timeout expires, the job request subscription and the status notifications associated to this job are discarded.

Valid values are in the range of 1 to 90. The default is 5 days. Changes to this parameter are effective immediately.

oslcAutomationDescription | ad

Description of the IBM Workload Scheduler automation service provider. Used in OSLC integration to register the IBM Workload Scheduler automation service provider in the Registry Services. This value is used to define a description for the service provider.

Changes to this parameter are effective immediately.

oslcAutomationTitle | at

Title of the IBM Workload Scheduler automation service provider. Used in OSLC integration to register the IBM Workload Scheduler automation service provider in the Registry Services. This value is used to uniquely identify the automation service provider. To easily identify the service provider you want to use, use a meaningful title for each IBM Workload Scheduler automation service provider registered in the same Registry Services.

Changes to this parameter are effective immediately.

oslcProviderUri | pu

Address of the IBM Workload Scheduler service provider. Used in OSLC integration to register the IBM Workload Scheduler service provider in the Registry Services. Use the format `https://hostname:port`, where `hostname` is the name of the host used to connect to the master domain manager. For example, `https://myProviderHostanme.com:31115`.

Changes to this parameter are effective immediately.

oslcProvisioningDescription | pd

Description of the IBM Workload Scheduler automation service provider. Used in OSLC integration to register the IBM Workload Scheduler automation service provider in the Registry Services. This value is used to define a description for the service provider.

Changes to this parameter are effective immediately.

oslcProvisioningTitle | pt

Title of the IBM Workload Scheduler provisioning service provider. Used in OSLC integration to register the IBM Workload Scheduler provisioning service provider in the Registry Services. This value is used to uniquely identify the provisioning service provider. To easily identify the service provider you want to use, use a meaningful title for each IBM Workload Scheduler provisioning service provider registered in the same Registry Services.

Changes to this parameter are effective immediately.

oslcRegistryPassword | rp

Password of the user connecting to the Registry Services. Used in OSLC integration to register the IBM Workload Scheduler service provider in the Registry Services.

Changes to this parameter are effective immediately.

oslcRegistryUri | cu

Address of the Registry Services. Used in OSLC integration to register the IBM Workload Scheduler service provider in the Registry Services. Use the format `https://hostname:port/oslc/pr`.

Changes to this parameter are effective immediately.

oslcRegistryUser | ru

User connecting to the Registry Services. Used in OSLC integration to register the IBM Workload Scheduler service provider in the Registry Services.

Changes to this parameter are effective immediately.

promotionOffset | po

Promotion offset. Used in workload service assurance. Specify when a job become eligible for promotion in terms of the number of seconds before its critical start time is reached. Applies only to jobs that are flagged as critical in a job stream definition and to their predecessor jobs. A critical job and its predecessors make up a critical network.

When a predecessor jeopardizes the timely completion of the critical job, it is *promoted*; that is, it is assigned additional resources and its submission is prioritized with respect to other jobs that are out of the critical network. Also critical jobs might be promoted.

The scheduler calculates the critical start time of a critical job by subtracting its estimated duration from its deadline. It calculates the critical start time of a critical predecessor by subtracting its estimated duration from the critical start time of its next successor. Within a critical network the scheduler calculates the critical start time of the critical job first and then works backwards along the chain of predecessors. These calculations are reiterated as many times as necessary until the critical job has run.

This option is only active if *enWorkloadServiceAssurance* is set to *yes*.

The default is 120 seconds.

Run JnextPlan to make this change effective.

resubmitJobName | rj

Name of the job in charge of resubmitting the job stream. Specify the name of the Job Stream Submission job which is automatically added to the plan to resubmit a new instance of the job stream where the start condition is defined.

The default value is *MASTERAGENTS#restart_StartCond*, where MASTERAGENTS is the name of the pool workstation on which the Job Stream Submission job runs. The maximum length for the workstation name

is 16 bytes, and the maximum length for the job name is 40 bytes. Changes to this parameter are effective immediately.

resubmitJobUserName | rw

Name of the user in charge of resubmitting the job stream. Specify the user name which owns the Job Stream Submission job. The Job Stream Submission job is automatically added to the plan to resubmit a new instance of the job stream where the start condition is defined.

The default value is *TWS_User*. Changes to this parameter are effective immediately. If the user defined in the **resubmitJobUserName** property does not exist, the user name and password defined on WebSphere Application Server Liberty Base installed on the master domain manager or backup master domain manager are used. This implies that the user defined in the **resubmitJobUserName** property must be the same both on the master domain manager and on the backup master domain manager, or must be changed immediately after switching the master domain manager.

riskConfidence | rc

Critical Jobs Risk Confidence Specifies when a critical job must be set as **High Risk**, comparing the confidence factor of completing before deadline and the percentage specified in this parameter. If the probability of completing before the deadline is below **riskConfidence**, then the critical job is considered at high risk. Valid values are in the range 1-99. The default value is 80% when you perform a fresh installation. If you upgrade a previous version to the current version, the default value is 50% for maintaining backward compatibility. This option is effective immediately.

sccdUrl | du

IBM SmartCloud Control Desk URL. Used in event rule management. If you use rules that implement an action that opens a ticket to an IBM SmartCloud Control Desk server (or any other application that can open ticket in IBM SmartCloud Control Desk format), specify the IBM SmartCloud Control Desk URL. You can change this value when you define the action if you want to use a different IBM SmartCloud Control Desk URL.

The default value is "`http://localhost:8080/maximo/oslc/os/oslcincident`". Changes to this parameter are effective immediately.

sccdUserName | dn

SmartCloud Control Desk user name. Used in event rule management. If you deploy rules that implement an action that opens a ticket by using the SmartCloud Control Desk, specify the identifier of the user connecting to the SmartCloud Control Desk server.

The default value is the IBM Workload Scheduler user on the master domain manager. Changes to this parameter are effective immediately.

sccdUserPassword | dp

SmartCloud Control Desk user password. Used in event rule management. If you deploy rules that implement an action that opens a ticket by using the SmartCloud Control Desk, specify the password associated with the user connecting to the SmartCloud Control Desk server. The password is stored in an encrypted form.

Changes to this parameter are effective immediately.

servicenowUrl | nu

ServiceNow URL. Used in event rule management. If you use rules that implement an action that opens an incident in ServiceNow (or any other application that can open an incident in the ServiceNow format), specify the ServiceNow URL. You can change this value when you define the action if you want to use a different ServiceNow URL.

The default value is "`http://localhost:8080/api/now/v1/table/incident`". Changes to this parameter are effective immediately.

servicenowUserName | nn

ServiceNow user name. Used in event rule management. If you use rules that implement an action that opens an incident in ServiceNow, specify the identifier of the user connecting to the ServiceNow server.

The default value is the IBM Workload Scheduler user on the master domain manager. Changes to this parameter are effective immediately.

servicenowUserPassword | np

ServiceNow user password. Used in event rule management. If you use rules that implement an action that opens an incident in ServiceNow, specify the password associated with the user connecting to the ServiceNow server.

Changes to this parameter are effective immediately.

smtpServerName | sn

SMTP server name. Used in event rule management. If you deploy rules implementing an action that sends emails via an SMTP server, specify the name of the SMTP server to be used by the mail plug-in.

The default value is *localhost*. Changes to this parameter are effective immediately.

smtpServerPort | sp

SMTP Server port. Used in event rule management. If you deploy rules implementing an action that sends emails via an SMTP server, specify the port number used to connect to the SMTP server by the mail plug-in. Valid values are in the range 0–65535.

The default value is 25. Changes to this parameter are effective for the next mail send action performed.

smtpUseAuthentication | ua

Mail plug-in uses SMTP authentication. Used in event rule management. If you deploy rules implementing an action that sends emails via an SMTP server, specify if the SMTP connection needs to be authenticated. Values are *yes* or *no*.

The default is *no*. Changes to this parameter are effective immediately.

smtpUserName | un

SMTP server user name. Used in event rule management. If you deploy rules implementing an action that sends emails via an SMTP server, specify the SMTP server user name.

The default value is the name of the IBM Workload Scheduler user (the <TWS_user>) on the master domain manager. Changes to this parameter are effective immediately.

smtpUserPassword | up

SMTP server user password. Used in event rule management. If you deploy rules implementing an action that sends emails via an SMTP server, specify the SMTP server user password. The password is stored in an encrypted form.

Changes to this parameter are effective immediately.

smtpUseSSL | us

Mail plug-in uses SSL. Used in event rule management. If you deploy rules implementing an action that sends emails via an SMTP server, specify if the SMTP connection is to be authenticated via SSL. Values are *yes* or *no*.

The default is *no*. Changes to this parameter are effective immediately.

smtpUseTLS | tl

Mail plug-in uses TLS protocol. Used in event rule management. If you deploy rules implementing an action that sends emails via an SMTP server, specify if the SMTP connection is to be authenticated via the Transport Layer Security (TLS) protocol. Values are *yes* or *no*.

The default is *no*. Changes to this parameter are effective immediately.

startOfDay | sd

Start time of processing day. Specify the start time of the IBM Workload Scheduler processing day in 24-hour format: *hhmm* (0000-2359).

The default value is 0000 (0:00 a.m.), but if you upgraded your environment to version 9.5 starting from a version earlier than version 8.6, the default value is 0600 (6:00 a.m.). If you change this option, you must also change the launch time of the *final* Job Scheduler, which is usually set to one minute before the start time. Run JnextPlan to make the change of *startOfDay* effective.

If you need to modify in a production environment the start of day and the FINAL job stream launch time, run the following command twice:

```
JnextPlan -to newDay newSoD TZ customerTimezone
```

For example, if your timezone is America/New_York and on the 30th of August you want to modify the start of day from 0000 to 0400, perform the following steps:

1. Modify the **startOfDay** global option using optman:

```
optman chg sd=0400
```

2. Modify the FINAL job stream launch modifying the **at** keyword from 2359 to 0359.

3. Run the following command twice:

```
JnextPlan -to 08/31/2022 0400 tz America/New_York
```

statsHistory | sh

Job statistics history period. Specify the number of days for which you want to maintain job statistics. Statistics are discarded on a FIFO (first-in first-out) basis. For example, if you leave the default value of *400*, statistics are maintained for the last 400 days. This has no effect on job standard list files, which must be removed with the *rmstdlist* command. See the *IBM Workload Scheduler: User's Guide and Reference* for information about the *rmstdlist* command.

The default value is *400*. Run *JnextPlan* to make this change effective in the plan. For the database, this option takes effect immediately.

startConditionDeadlineOffset | cd

Start condition deadline offset. The default offset set for the start condition deadline in 24 hour format: "hhmm" (0001-9959). Specify the time range during which the start condition is active.

The default value is *2400* and the range is *0001 - 9959*. Changes to this parameter are effective immediately.

TECServerName | th

EIF Probe server name. Used in event rule management. If you use rules implementing an action that forwards events to a Tivoli Enterprise Console® or Tivoli Business Service Manager server (or any other application that processes events in TEC or TBSM format), specify the EIF Probe server name. If you want to use a different EIF Probe server, you can change this value when you define the action.

The default is *localhost*. Run *JnextPlan* to make this change effective.

TECServerPort | tp

EIF Probe server port. Used in event rule management. If you use rules implementing an action that forwards events to a Tivoli Enterprise Console® or Tivoli Business Service Manager server (or any other application that processes events in TEC or TBSM format), specify the port number of the EIF Probe server. If you want to use a different EIF Probe server, you can change this value when you define the action.

The default port number is *5529*. Run *JnextPlan* to make this change effective.

untilDays | ud

Remove obsolete job and job stream instances from the plan. If an **until** time (latest start time) has not been specified for a job or job stream, then the default **until** time is calculated adding the value of this option, expressed in number of days, to the scheduled time of the job or job stream. If the *enCarryForward* option is set to **all**, and the number of days specified for *untilDays* is reached, then any job or job stream instance in the plan that ended in error is automatically removed from the plan and not added to the new production plan.

The default value is **0**. If the default value is used, then for jobs, no default time is set for the **until** time (latest start time) . For job streams, if the default is used, then the default until time is 2 days.

Run JnextPlan to make this change effective.

workstationLimit | wl

The workstation limit.

Used in the automatic dynamic agent registration. This parameter specifies the dynamic agent workstation limit value that the dynamic agent workstation assumes after the workstation is added to the plan. You can later modify the dynamic agent workstation limit value by using the conman command line or the Dynamic Workload Console.

Valid values are in the *0-1024* range.

The default is *100*. Changes to this parameter are effective immediately.

zOSRemoteServerName | zr

IBM Z Workload Scheduler connector remote server name. Used in event rule management. If you deploy rules implementing an action that submits job streams to the IBM Z Workload Scheduler controller, enter the name of the controller specified as the engine to the Z connector. It must exactly match the Z connector engine name and is case sensitive.

After changing the value of this parameter, the change becomes effective when the next `submit` action is run.

zOSServerName | zs

IBM Z Workload Scheduler connector server name. Used in event rule management. If you deploy rules implementing an action that submits job streams to the IBM Z Workload Scheduler controller, specify the name or the hostname of the system where the IBM Z Workload Scheduler connector runs. The default value is `localhost`.

After changing the value of this parameter, the change becomes effective when the next `submit` action is run.

zOSServerPort | zp

IBM Z Workload Scheduler connector server port. Used in event rule management. If you deploy rules implementing an action that submits job streams to the IBM Z Workload Scheduler controller, specify the bootstrap port number of the IBM Z Workload Scheduler connector server. Valid values are in the range 0-65535. The default value is 31217.

After changing the value of this parameter, the change becomes effective when the next `submit` action is run.

zOSUserName | zu

IBM Z Workload Scheduler connector user name. Used in event rule management. If you deploy rules implementing an action that submits job streams to the IBM Z Workload Scheduler controller, specify the IBM Z Workload Scheduler connector user name required to access the IBM Z Workload Scheduler engine.

After changing the value of this parameter, the change becomes effective when the next `submit` action is run.

zOSUserPassword | zw

IBM Z Workload Scheduler connector user password. Used in event rule management. If you deploy rules implementing an action that submits job streams to the IBM Z Workload Scheduler controller, specify the IBM Z Workload Scheduler connector user password required to access the IBM Z Workload Scheduler engine. The password is stored in encrypted form.

After changing the value of this parameter, the change becomes effective when the next `submit` action is run.

Setting local options

Set local options, such as general attributes of the workstation for the IBM Workload Scheduler processes, in the `localopts` file. Changes do not take effect until `netman` is stopped (**conman shut;wait**) and restarted (**StartUp**).

During the installation process, a working copy of the local options file is installed as `TWA_DATA_DIR/localopts`.

The `localopts` file is not modified during the agent upgrade process. The file generated by the upgrade process is saved to the `/config` directory, to maintain your custom values, if any. You can then merge the two files with your customized values and save the resulting file in the `TWA_DATA_DIR` folder.

A template file containing default settings is located in `TWA_DATA_DIR`.



Note: All of the SSL settings in the `localopts` file relate to the network communications and do not relate to the Dynamic Workload Console.

The options in the `localopts` file are described in the following sections:

- [Localopts summary on page 51](#)
- [Localopts details on page 55](#)
- [Local options file example on page 72](#)

Localopts summary

General attributes of the workstation:

thiscpu = *workstation*

merge stdlists = *yes/no*

stdlist width = *columns*

syslog local = *facility*

restricted stdlists = *yes/no*

The attributes of the workstation for the batchman process:

bm check file = *seconds*

bm check status = *seconds*

bm look = *seconds*

bm read = *seconds*

bm stats = *on/off*

bm verbose = *on/off*

bm check until = *seconds*

bm check deadline = *seconds*

bm late every = *minutes*

The attributes of the workstation for the jobman process:

jm interactive old = *yes/no*

jm job table size = *entries*

jm load user profile = *on/off*

jm look = *seconds*

jm nice = *value*

jm promoted nice = *UNIX® and Linux® critical job priority*

jm promoted priority = *Windows® critical job priority*

jm no root = *yes/no*

jm file no root = *yes/no*

jm read = *seconds*

The attributes of the workstation for the mailman process:

mm planoffset = *HHMM*

mm response = *seconds*

mm retrylink = *seconds*

mm sound off = *yes/no*

mm unlink = *seconds*

mm cache mailbox = *yes/no*

mm cache size = *bytes*

mm resolve master = *yes/no*

autostart monman = *yes/no*

mm read = *minutes*

The attributes of the workstation for the netman process:

nm mortal = *yes/no*

nm port = *port number*

nm read = *seconds*

nm retry = *seconds*

The attributes of the workstation for the writer process:

wr read = *seconds*

wr unlink = *seconds*

wr enable compression = *yes/no*

Optional attributes of the workstation for remote database files

mozart directory = *mozart_share*
parameters directory = *parms_share*
unison network directory = *unison_share*

The attributes of the workstation for the custom formats

date format = *integer*
composer prompt = *key*
conman prompt = *key*
switch sym prompt = *key*

The attributes of the workstation for the customization of I/O on mailbox files

sync level = *low|medium|high*

The attributes of the workstation for networking

tcp timeout = *seconds*
tcp connect timeout = *seconds*

The attributes of the workstation for SSL - General

ssl auth mode = *caonly|string|cpu*
ssl auth string = *string*
ssl fips enabled = *yes/no*
nm ssl full port = *value*
nm ssl port = *value*

OpenSSL attributes of the workstation - only used if *ssl fips enabled* = "no"

ssl key = **.pem*
ssl certificate = **.pem*
ssl key pwd = **.sth*
ssl ca certificate = **.crt*
ssl random seed = **.rnd*
ssl encryption cipher = *cipher*
cli ssl server auth = *yes|no*
cli ssl cipher = *string*
cli ssl server certificate = *file_name*
cli ssl trusted dir = *directory_name*
cli ssl tls10 cipher = *HIGH|cipher*
cli ssl tls11 cipher = *HIGH|cipher*
cli ssl tls12 cipher = *HIGH|cipher*
ssl tls10 cipher = *HIGH|cipher*
ssl tls11 cipher = *HIGH|cipher*
ssl tls12 cipher = *HIGH|cipher*

GSKit attributes of the workstation - only used if *ssl fips enabled = "yes"*

ssl keystore file = *.kdb
ssl certificate keystore label = name
ssl keystore pwd = *.sth
cli ssl keystore file = *.kdb
cli ssl certificate keystore label = name
cli ssl keystore pwd = *.sth
cli gsk tls10 cipher = DFLT|cipher
cli gsk tls11 cipher = DFLT|cipher
cli gsk tls12 cipher = DFLT|cipher
gsk tls10 cipher = DFLT|cipher
gsk tls11 cipher = DFLT|cipher
gsk tls12 cipher = DFLT|cipher

The attributes of the workstation for the WebSphere Application Server Liberty Base

local was = yes|no

Application server check attributes on the workstation

appserver check interval = minutes
appserver auto restart = on|off
appserver min restart time = minutes
appserver max restarts = number
appserver count reset interval = hours
appserver service name = name

The IBM Workload Scheduler instance is a command line client

is remote cli = yes|no

Attributes for CLI connections

host = host_name
protocol = protocol
port = port number
proxy = proxy server
proxy port = proxy server port number
time out = seconds
followlocation= true|false
defaultws = master_workstation
useropts = useropts_file



Note:



1. The SSL attributes for the command line client connection will depend on which SSL method is in use. They are included in the relevant section and all commence with "cli".
2. The command lines for the dynamic domain manager and backup dynamic domain manager will work only if you configure the **host** and **port** attributes.

Event Management parameters

can be event processor = *yes|no*

er load = *yes|no*

Centralized Agent Update parameters

DownloadDir = *directory_name*

Current Folder

current folder = */foldername>*

Encryption options

encryptkeystorefile = *"/opt/IBM/TWA/TWSDATA/ssl/aes/<key_file>"*

encryptkeystorepwd = *"/opt/IBM/TWA/TWSDATA/ssl/aes/<keystore_password>"*

encryptlabel = *"default"*

#decryptlabellist =



Note: The `localopts` file syntax is not case-sensitive, and the spaces between words in the option names are ignored. For example, you can validly write **is remote cli** as:

- is remote cli
- Is Remote CLI
- isremotecli
- ISREMOTECli
- isRemoteCLI
- ...

Localopts details

comment

Treats everything from the indicated character (#) to the end of the line as a comment.

appserver auto restart = yes|no

Requests the `appservman` process to automatically start WebSphere Application Server Liberty Base if it is found down. The default is `yes`.

appserver check interval = *minutes*

Specifies the frequency in minutes that the `appservman` process is to check that WebSphere Application Server Liberty Base is still running. The default is 3 minutes.

appserver count reset interval = *hours*

Specifies the time interval in hours after which the restart count is reset from the last WebSphere Application Server Liberty Base start. The default is 24 hours.

appserver max restarts = *number*

Specifies the maximum number of restarting attempts the `appservman` process can make before giving up and exiting without restarting WebSphere Application Server Liberty Base. The counter is reset if WebSphere Application Server Liberty Base runs for longer than the `appserver count reset interval` value. The default is 5.

appserver min restart time = *minutes*

Specifies in minutes the minimum elapsed time the `appservman` process must wait between each attempt to restart the WebSphere Application Server Liberty Base if it is down. If this value is less than the `appserver check interval`, the WebSphere Application Server Liberty Base is restarted as soon as it is found down. If it is found down before this time interval (min restart time) has elapsed, `appservman` exits without restarting it. The default is 2 minutes.

appserver service name = *name*

Only in Windows® environments. Specifies the name of the WebSphere Application Server Liberty Base windows service if different from the standard name. This field is generally not used.

autostart monman = *yes|no*

Used in event rule management. Restarts the monitoring engine automatically when the next production plan is activated (on Windows® also when IBM Workload Scheduler is restarted). The default is `yes`.

bm check deadline = *seconds*

Specify the minimum number of seconds Batchman waits before checking if a job has missed its deadline. The check is performed on all jobs and job streams included in the Symphony file, regardless of the workstation where the jobs and job streams are defined. Jobs and job streams with expired deadlines are marked as late in the local Symphony file. To obtain up-to-date information about the whole environment, define this option on the master domain manager. Deadlines for critical jobs are evaluated automatically, independently of the **bm check deadline** option. To disable the option and not check deadlines, enter a value of zero, the default value.

bm check file = *seconds*

Specify the minimum number of seconds Batchman waits before checking for the existence of a file that is used as a dependency. The default is 120 seconds.

bm check status = *seconds*

Specify the number of seconds Batchman waits between checking the status of an internetwork dependency. The default is 300 seconds.

bm check until = seconds

Specify the maximum number of seconds Batchman waits before reporting the expiration of an Until time for job or Job Scheduler. Specifying a value below the default setting (300) might overload the system. If it is set below the value of Local Option **bm read**, the value of **bm read** is used in its place. The default is 300 seconds.

bm look = seconds

Specify the minimum number of seconds Batchman waits before scanning and updating its production control file. If you install the 9.4, FP1 version as a fresh installation, the default value is automatically set to 5 for improving product performance. The previous default value was 15 seconds and is maintained if you perform a product upgrade.

bm read = seconds

Specify the maximum number of seconds Batchman waits for a message in the `intercom.msg` message file. If no messages are in queue, Batchman waits until the timeout expires or until a message is written to the file. If you install the 9.4, FP1 version as a fresh installation, the default value is automatically set to 3 for improving product performance. The previous default value was 10 seconds and is maintained if you perform a product upgrade.

bm stats = on|off

To have Batchman send its startup and shut down statistics to its standard list file, specify **on**. To prevent Batchman statistics from being sent to its standard list file, specify **off**. The default is **off**.

bm verbose = on|off

To have Batchman send all job status messages to its standard list file, specify **on**. To prevent the extended set of job status messages from being sent to the standard list file, specify **off**. The default is **off**.

bm late every = minutes

When an **every** job does not start at its expected start time, **bm late every** specifies the maximum number of minutes that elapse before IBM Workload Scheduler skips the job. This option applies only to jobs defined with **every** option together with the **at** time dependency, it has no impact on jobs that have only the **every** option.

can be event processor = yes|no

Specify if this workstation can act as event processing server or not. It is set by default to **yes** for master domain managers and backup masters, otherwise it is set to **no**.

cli gsk tls10 cipher=DFLT|cipher

Only used if SSL is defined using GSKit (`ssl fips enabled="yes"`). Specify the cipher to be used with the TLS 1.0 protocol in association with GSKit when using the IBM Workload Scheduler command line. Restart the agent to make the changes effective. This keyword is optional and must be manually inserted in the localopts file. When specified, it overrides the default option. If you set more parameters with different versions of the same protocol, the protocol with the lowest version is used.

cli gsk tls11 cipher=DFLT|cipher

Only used if SSL is defined using GSKit (`ssl fips enabled="yes"`). Specify the cipher to be used with the TLS 1.1 protocol in association with GSKit when using the IBM Workload Scheduler command line. Restart the agent

to make the changes effective. This keyword is optional and must be manually inserted in the localopts file. When specified, it overrides the default option. If you set more parameters with different versions of the same protocol, the protocol with the lowest version is used.

cli gsk tls12 cipher=DFLT|cipher

Only used if SSL is defined using GSKit (`ssl fips enabled="yes"`). Specify the cipher to be used with the TLS 1.2 protocol in association with GSKit when using the IBM Workload Scheduler command line. Restart the agent to make the changes effective. This keyword is optional and must be manually inserted in the localopts file. When specified, it overrides the default option. If you set more parameters with different versions of the same protocol, the protocol with the lowest version is used.

cli ssl tls10 cipher=HIGH|cipher

Only used if SSL is defined using OpenSSL (`ssl fips enabled="no"`). Specify the cipher to be used with the TLS 1.0 protocol in association with SSL when using the IBM Workload Scheduler command line. Restart the agent to make the changes effective. This keyword is optional and must be manually inserted in the localopts file. When specified, it overrides the default option. If you set more parameters with different versions of the same protocol, the protocol with the lowest version is used.

cli ssl tls11 cipher=HIGH|cipher

Only used if SSL is defined using OpenSSL (`ssl fips enabled="no"`). Specify the cipher to be used with the TLS 1.1 protocol in association with SSL when using the IBM Workload Scheduler command line. Restart the agent to make the changes effective. This keyword is optional and must be manually inserted in the localopts file. When specified, it overrides the default option. If you set more parameters with different versions of the same protocol, the protocol with the lowest version is used.

cli ssl tls12 cipher=HIGH|cipher

Only used if SSL is defined using OpenSSL (`ssl fips enabled="no"`). Specify the cipher to be used with the TLS 1.2 protocol in association with SSL when using the IBM Workload Scheduler command line. Restart the agent to make the changes effective. This keyword is optional and must be manually inserted in the localopts file. When specified, it overrides the default option. If you set more parameters with different versions of the same protocol, the protocol with the lowest version is used.

gsk tls10 cipher=DFLT|cipher

Only used if SSL is defined using GSKit (`ssl fips enabled="yes"`). Specify the cipher to be used with the TLS 1.0 protocol in association with GSKit. Restart the agent to make the changes effective. This keyword is optional and must be manually inserted in the localopts file. When specified, it overrides the default option. If you set more parameters with different versions of the same protocol, the protocol with the lowest version is used.

gsk tls11 cipher=DFLT|cipher

Only used if SSL is defined using GSKit (`ssl fips enabled="yes"`). Specify the cipher to be used with the TLS 1.1 protocol in association with GSKit. Restart the agent to make the changes effective. This keyword is optional and must be manually inserted in the localopts file. When specified, it overrides the default option. If you set more parameters with different versions of the same protocol, the protocol with the lowest version is used.

gsk tls12 cipher=DFLT|cipher

Only used if SSL is defined using GSKit (`ssl fips enabled="yes"`). Specify the cipher to be used with the TLS 1.2 protocol in association with GSKit. Restart the agent to make the changes effective. This keyword is optional and must be manually inserted in the localopts file. When specified, it overrides the default option. If you set more parameters with different versions of the same protocol, the protocol with the lowest version is used.

ssl tls10 cipher=HIGH|cipher

Only used if SSL is defined using OpenSSL (`ssl fips enabled="no"`). Specify the cipher to be used with the TLS 1.0 protocol in association with SSL. Restart the agent to make the changes effective. This keyword is optional and must be manually inserted in the localopts file. If you set more parameters with different versions of the same protocol, the protocol with the lowest version is used.

ssl tls11 cipher=HIGH|cipher

Only used if SSL is defined using OpenSSL (`ssl fips enabled="no"`). Specify the cipher to be used with the TLS 1.1 protocol in association with SSL. Restart the agent to make the changes effective. This keyword is optional and must be manually inserted in the localopts file. If you set more parameters with different versions of the same protocol, the protocol with the lowest version is used.

ssl tls12 cipher=HIGH|cipher

Only used if SSL is defined using OpenSSL (`ssl fips enabled="no"`). Specify the cipher to be used with the TLS 1.2 protocol in association with SSL. Restart the agent to make the changes effective. This keyword is optional and must be manually inserted in the localopts file. If you set more parameters with different versions of the same protocol, the protocol with the lowest version is used.

cli ssl certificate keystore label = *string*

Only used if SSL is defined using GSKit (`ssl fips enabled="yes"`). Supply the label which identifies the certificate in the keystore when the command-line client is using SSL authentication to communicate with the master domain manager. The default is `IBM TWS 9.5 workstation`, which is the value of the certificate distributed with the product to all customers. This certificate is thus not secure and should be replaced with your own secure certificate. See [Configuring the SSL connection protocol for the network on page 323](#).

cli ssl keystore file = *file_name*

Only used if SSL is defined using GSKit (`ssl fips enabled="yes"`). Specify the name of the keystore file used for SSL authentication when the command-line client is using SSL authentication to communicate with the master domain manager. The default is `TWA_home/TWS/ssl/TWSPublicKeyFile.pem`. This file is part of the SSL configuration distributed with the product to all customers. It is thus not secure and should be replaced with your own secure SSL configuration. See [Configuring the SSL connection protocol for the network on page 323](#).

cli ssl keystore pwd = *file_name*

Only used if SSL is defined using GSKit (`ssl fips enabled="yes"`). Specify the password file of the keystore used for SSL authentication when the command-line client is using SSL authentication to communicate with the master domain manager. This file is part of the SSL configuration distributed with the product to

all customers. It is thus not secure and should be replaced with your own secure SSL configuration. See [Configuring the SSL connection protocol for the network on page 323](#).

cli ssl cipher = *cipher_class*

Only used if SSL is defined using OpenSSL (`ssl fips enabled="no"`) Specify the cipher class to be used when the command-line client and the server are using SSL authentication. Use one of the common cipher classes listed in [Table 20: Valid encryption cipher classes on page 60](#). The default is MD5.

If you want to use an OpenSSL cipher class not listed in the table, use the following command to determine if your required class is supported:

```
openssl ciphers class_name
```

where *class_name* is the name of the class you want to use. If the command returns a cipher string, the class can be used.

Table 20. Valid encryption cipher classes

Encryption cipher class	Description
SSLv3	SSL version 3.0
TLS	Only for IBM® Workload Scheduler, version 9.3. users: before enabling SSL communication, manually modify this value to TLSv1 , as described in Setting local options on page 51 . For users with V9.3 Fix Pack 1 or later, no manual intervention is required. The default value is TLSv1.
TLSv1	TLS version 1.0
EXP	Export
EXPORT40	40-bit export
MD5	Ciphers using the MD5 digest, digital signature, one-way encryption, hash or checksum algorithm.
LOW	Low strength (no export, single DES)
MEDIUM	Ciphers with 128 bit encryption
HIGH	Ciphers using Triple-DES
NULL	Ciphers using no encryption

cli ssl server auth = yes|no

Only used if SSL is defined using OpenSSL (`ssl fips enabled="no"`) Specify **yes** if server authentication is to be used in SSL communications with the command line client. The default is **no**.

cli ssl server certificate = *file_name*

Only used if SSL is defined using OpenSSL (`ssl fips enabled="no"`) Specify the file, including its full directory path, that contains the SSL certificate when the command-line client and the server use SSL authentication

in their communication. There is no default. See [Configuring the SSL connection protocol for the network on page 323](#).

cli ssl trusted dir = *directory_name*

Only used if SSL is defined using OpenSSL (`ssl fips enabled="no"`) Specify the directory that contains an SSL trusted certificate contained in files with hash naming (#) when the command-line client and the server are using SSL authentication in their communication. When the directory path contains blanks, enclose it in double quotation marks (""). There is no default.

composer prompt = *prompt*

Specify the prompt for the composer command line. The prompt can be of up to 10 characters in length. The default is dash (-).

conman prompt = *prompt*

Specify the prompt for the conman command line. The prompt can be of up to 8 characters in length. The default is percent (%).

current folder = */foldername>*

When submitting commands that involve folders from either the composer or conman command line, you can change the default folder or working directory from the root (/) to another folder path so that you can submit commands from the composer or conman command line using relative folder paths.

date format = 0|1|2|3

Specify the value that corresponds to the date format you require. The values can be:

- 0 corresponds to *yy/mm/dd*
- 1 corresponds to *mm/dd/yy*
- 2 corresponds to *dd/mm/yy*
- 3 indicates usage of Native Language Support variables

The default is 1.

followlocation

Set this property to `true` to enable the HTTP protocol. You cannot enable the HTTP protocol from the command line. This property instructs the composer command to follow any 'Location: header' that the server sends as part of the HTTP header in a 3xx response. The 'Location: header' can specify a relative or an absolute URL to follow.

defaultws = *manager_workstation*

The default workstation when you are accessing using a command line client. Specify the domain manager workstation.

DownloadDir = *directory_name*

Defines the name of the directory where the fix pack installation package or upgrade elmage is downloaded during the centralized agent update process. If not specified, the following default directory is used:

On Windows operating systems:

`TWA_home\TWS\stdlist\JM\download`

On UNIX operating systems:

`TWA_home/TWS/stdlist/JM/download`

er load = yes|no

For UNIX and Linux operating systems only. If set to **yes**, specifies that the IBM user profile should be loaded when running a GenericAction EventRule. The default value is **no**.

host = hostname_or_IP_address

The name or IP address of the host when accessing using a command line client. For **Agents**, the host or ip address of the master is used. For **Backup Master Domain Manager** the value is the default: 127.0.0.1

is remote cli = yes|no

Specify if this instance of IBM Workload Scheduler is installed as a command line client (yes).

jm interactive old = yes|no

Only for Windows operating systems starting from Vista and later versions. To comply with security restrictions introduced with the Vista version of Windows operating systems, only for fault-tolerant agents, IBM Workload Scheduler runs interactive jobs only if the `streamlogon` user has a valid, interactive session. Specify **yes** to allow jobman to start interactive jobs even if there are no active sessions for the `streamlogon` user. Specify **no** to allow jobman to start interactive jobs only if there are active sessions for the `streamlogon` user. The default is **no**.

jm job table size = entries

Specify the size, in number of entries, of the job table used by Jobman. The default is 1024 entries.

jm load user profile = on|off

Only on Windows operating systems. Specify if the jobman process loads the user profile and its environment variables for the user specified in the logon field of each job, before starting the job on the workstation. Specify **on** to load the user profile on the workstation before running jobs for the logon user; otherwise specify **off**. Roaming profiles are not supported. The default is **on**.

jm look = seconds

Specify the minimum number of seconds Jobman waits before looking for completed jobs and performing general job management tasks. The default is 300 seconds.

jm nice = nice_value

For UNIX® and Linux® operating systems only, specify the **nice** value to be applied to jobs launched by Jobman to change their priority in the kernel's scheduler. The default is zero.

The **nice** boundary values vary depending upon each specific platform, but generally lower values correspond to higher priority levels and vice versa. The default depends upon the operating system.

Applies to jobs scheduled by the root user only. Jobs submitted by any other user inherit the same **nice** value of the Jobman process.

See also [jm promoted nice on page 63](#).

jm file no root = yes|no

For UNIX® and Linux® operating systems only, specify **yes** to prevent Jobman from executing commands in file dependencies as **root**. Specify **no** to allow Jobman to execute commands in file dependencies as **root**. The default is **no**.

jm no root = yes|no

For UNIX® and Linux® operating systems only, specify **yes** to prevent Jobman from launching **root** jobs. Specify **no** to allow Jobman to launch **root** jobs. The default is **yes**.

jm promoted nice = nice_value

Used in workload service assurance. For UNIX® and Linux® operating systems only, assigns the priority value to a critical job that needs to be promoted so that the operating system processes it before others. Applies to critical jobs or predecessors that need to be promoted so that they can start at their critical start time.

Boundary values vary depending upon each specific platform, but generally lower values correspond to higher priority levels and vice versa. The default is -1.

Be aware that:

- The promotion process is effective with negative values only. If you set a positive value, the system runs it with the -1 default value and logs a warning message every time Jobman starts.
- An out of range value (for example -200), prompts the operating system to automatically promote the jobs with the lowest allowed **nice** value. Note that in this case no warning is logged.
- Overusing the promotion mechanism (that is, defining an exceedingly high number of jobs as mission critical and setting the highest priority value here) might overload the operating system, negatively impacting the overall performance of the workstation.

You can use this and the [jm nice on page 62](#) options together. If you do, remember that, while **jm nice** applies only to jobs submitted by the root user, **jm promoted nice** applies only to jobs that have a critical start time. When a job matches both conditions, the values set for the two options add up. For example, if on a particular agent the local options file has:

```
jm nice= -2
jm promoted nice= -4
```

when a critical job submitted by the root user needs to be promoted, it is assigned a cumulative priority value of -6.

jm promoted priority = value

Used in workload service assurance. For Windows® operating systems only, sets to this value the priority by which the operating system processes a critical job when it is promoted.

Applies to critical jobs or predecessors that need to be promoted so that they can start at their critical start time.

The possible values are:

- High
- AboveNormal (the default)
- Normal
- BelowNormal
- Low Or Idle

Note that if you set a lower priority value than the one non-critical jobs might be assigned, no warning is given and no mechanism like the one available for **jm promoted nice** sets it back to the default.

jm read = seconds

Specify the maximum number of seconds Jobman waits for a message in the `courier.msg` message file. The default is 10 seconds.

local was = yes|no

For master domain managers and backup masters connected to the IBM Workload Scheduler database. If set to **yes**, it improves the performance of Job Scheduler and job submission from the database. The default is **no**.

merge stdlists = yes|no

Specify **yes** to have all of the IBM Workload Scheduler control processes, except Netman, send their console messages to a single standard list file. The file is given the name **TWSmerge**. Specify **no** to have the processes send messages to separate standard list files. The default is **yes**.

mm cache mailbox = yes|no

Use this option to enable Mailman to use a reading cache for incoming messages. In this case, only messages considered essential for network consistency are cached. The default is **yes**.

mm cache size = messages

Specify this option if you also use **mm cache mailbox**. The maximum value (default) is **512**.

mm planoffset = HHMM

HHMM is an amount of time in the format hours and minutes. When IBM Workload Scheduler starts, this amount of time is used as an offset to check the Symphony plan validity according to this formula:

```
current_timestamp < (Symphony_end_timestamp - HHMM)
```

If the result is true, that is, the current time is earlier than the Symphony planned end time minus the offset, the Symphony plan is considered valid and IBM Workload Scheduler starts. If the result is false, IBM Workload Scheduler does not start and an error is logged. The default for this optional attribute is an empty value; in this case, no check is performed by IBM Workload Scheduler on the validity of the plan. This check might be necessary when a domain manager stops because of an unplanned outage and restarts later, when a new domain manager has been started in the meanwhile, because not all the correct recovery procedures were run

to exclude it from the IBM Workload Scheduler network. As a consequence, there are two domain managers running at the same time on the same fault-tolerant agent creating scheduling issues on all the fault-tolerant agents.

mm read = seconds

Specify the maximum number of seconds Mailman waits for a connection with a remote workstation. The default is 15 seconds.

mm resolve master = yes|no

When set to **yes** the \$MASTER variable is resolved at the beginning of the production day. The host of any extended agent is switched after the next JnextPlan (long term switch). When it is set to **no**, the \$MASTER variable is not resolved at JnextPlan and the host of any extended agent can be switched after a conman **switchmgr** command (short- and long-term switch). Starting from Version 9.5 Fix Pack 2, the default is **no** (for previous releases, it was set to **yes**. When you switch a master domain manager and the original has mm resolve master set to **no** and the backup has mm resolve master set to **yes**, after the switch any extended agent that is hosted by \$MASTER switches to the backup master domain manager. When the backup master domain manager restarts, the keyword \$MASTER is locally expanded by Mailman. You should keep the mm resolve master value the same for master domain managers and backup domain managers.

mm response = seconds

Specify the maximum number of seconds Mailman waits for a response before reporting that a workstation is not responding. The minimum wait time for a response is **90** seconds. The default is 600 seconds.

mm retrylink = seconds

Specify the maximum number of seconds Mailman waits after unlinking from a non-responding workstation before it attempts to link to the workstation again. The default is 600 seconds. The **tomservers** optional mailman servers do not unlink non-responding agents. The link is repetitively checked every 60 seconds, which is the default **retrylink** for these servers.

mm sound off = yes|no

Specify how Mailman responds to a conman **telop ?** command. Specify **yes** to have Mailman display information about every task it is performing. Specify **no** to have Mailman send only its own status. The default is **no**.

mm symphony download timeout = seconds

Specify the maximum number of minutes Mailman waits after attempting to initialize a workstation on a slow network. If the timeout expires without the workstation being initialized successfully, Mailman initializes the next workstation in the sequence. The default is no timeout (0).

mm unlink = seconds

Specify the maximum number of seconds Mailman waits before unlinking from a workstation that is not responding. The wait time should not be less than the response time specified for the Local Option **nm response**. The default is 960 seconds.

nm mortal = yes|no

Specify **yes** to have Netman quit when all of its child processes have stopped. Specify **no** to have Netman keep running even after its child processes have stopped. The default is **no**.

nm port = port

Specify the TCP port number that Netman responds to on the local computer. This must match the TCP/IP port in the computer's workstation definition. It must be an unsigned 16-bit value in the range 1- 65535 (values between 0 and 1023 are reserved for services such as, FTP, TELNET, HTTP, and so on). The default is the value supplied during the product installation.

If you run event-driven workload automation and you have a security firewall, make sure this port is open for incoming and outgoing connections.

nm read = seconds

Specify the maximum number of seconds Netman waits for a connection request before checking its message queue for **stop** and **start** commands. The default is 10 seconds.

nm retry = seconds

Specify the maximum number of seconds Netman waits before retrying a connection that failed. The default is 800 seconds.

nm ssl full port = port

The port used to listen for incoming SSL connections when full SSL is configured by setting global option `enSSLFullConnection` to `yes` (see [Configuring full SSL security on page 324](#) for more details). This value must match the one defined in the `secureaddr` attribute in the workstation definition in the database. It must be different from the `nm port` local option that defines the port used for normal communication.

**Note:**

1. If you install multiple instances of IBM Workload Scheduler on the same computer, set all SSL ports to different values.
2. If you plan not to use SSL, set the value to 0.

There is no default.

nm ssl port = port

The port used to listen for incoming SSL connections, when full SSL is not configured (see [Configuring full SSL security on page 324](#) for more details). This value must match the one defined in the `secureaddr` attribute in the workstation definition in the database. It must be different from the `nm port` local option that defines the port used for normal communication.

**Note:**



1. If you install multiple instances of IBM Workload Scheduler on the same computer, set all SSL ports to different values.
2. If you plan not to use SSL, set the value to 0.

The default value is 31113.

port = port

The TCP/IP port number of the protocol used when accessing using a command line client. The default is 31116.

protocol = http|https

The protocol used to connect to the host when accessing using a command line client.

proxy = proxy_server_hostname_or_IP_address

The name of the proxy server used when accessing using a command line client.

proxy port = proxy_server_port

The TCP/IP port number of the proxy server used when accessing using a command line client.

restricted stdlists = yes|no

Use this option to set a higher degree of security to the `stdlist` directory (and to its subdirectories) allowing only selected users to create, modify, or read files.

This option is available for UNIX workstations only. After you define it, make sure you erase your current `stdlist` directory (and subdirectories) and that you restart IBM Workload Scheduler. The default is `no`.

If the option is not present or if it is set to `no`, the newly created `stdlist` directory and its subdirectories are unaffected and their rights are as follows:

```
drwxrwxr-x  2 twsmdm staff      4096 Nov  9 12:12
drwxrwxr-x   2 twsmdm staff      256 Nov  9 11:40 2009.11.09
drwxrwxr-x   2 twsmdm staff      4096 Nov  9 11:40 logs
drwxr-xr-x   2 twsmdm staff      4096 Nov  9 11:40 traces
```

If the option is set to `yes`, these directories have the following rights:

```
drwxr-x--x  5 twsmdm staff      256 Nov 13 18:15
rwxr-x--x   2 twsmdm staff      256 Nov 13 18:15 2009.11.13
rwxr-x--x   2 twsmdm staff      256 Nov 13 18:15 logs
rwxr-x--x   2 twsmdm staff      256 Nov 13 18:15 traces
```

Do the following to define and activate this option:

1. Change the line `restricted stdlists = no` to `restricted stdlists = yes` in your local options file.
2. Stop IBM Workload Scheduler.
3. Stop WebSphere Application Server Liberty Base if present.
4. Remove the `stdlist` directory (or at least its files and subdirectories).

5. Start IBM Workload Scheduler.
6. Start WebSphere Application Server Liberty Base if present.

ssl auth mode = caonly|string|cpu

The behavior of IBM Workload Scheduler during an SSL handshake is based on the value of the SSL authentication mode option as follows:

caonly

IBM Workload Scheduler checks the validity of the certificate and verifies that the peer certificate has been issued by a recognized CA. Information contained in the certificate is not examined. The default value.

string

IBM Workload Scheduler checks the validity of the certificate and verifies that the peer certificate has been issued by a recognized CA. It also verifies that the Common Name (CN) of the Certificate Subject matches the string specified into the SSL auth string option. See [ssl auth string = string on page 68](#).

cpu

IBM Workload Scheduler checks the validity of the certificate and verifies that the peer certificate has been issued by a recognized CA. It also verifies that the Common Name (CN) of the Certificate Subject matches the name of the workstation that requested the service.

ssl auth string = string

Used in conjunction with the **SSL auth mode** option when the "string" value is specified. The **SSL auth string** (ranges from 1 - 64 characters) is used to verify the certificate validity. The default string is "twS".

ssl ca certificate = file_name

Only used if SSL is defined using OpenSSL (`ssl fips enabled="no"`) Specify the name of the file containing the trusted certification authority (CA) certificates required for SSL authentication. The CAs in this file are also used to build the list of acceptable client CAs passed to the client when the server side of the connection requests a client certificate. This file is the concatenation, in order of preference, of the various PEM-encoded CA certificate files.

The default is `TWA_home/TWS/ssl/TWSTrustedCA.crt`. This file is part of the SSL configuration distributed with the product to all customers. It is thus not secure and should be replaced with your own secure SSL configuration. See [Configuring the SSL connection protocol for the network on page 323](#).

ssl certificate = file_name

Only used if SSL is defined using OpenSSL (`ssl fips enabled="no"`) Specify the name of the local certificate file used in SSL communication.

The default is `TWA_home/TWS/ssl/TWSPublicKeyFile.pem`. This file is part of the SSL configuration distributed with the product to all customers. It is thus not secure and should be replaced with your own secure SSL configuration. See [Configuring the SSL connection protocol for the network on page 323](#).

ssl certificate keystore label = *string*

Only used if SSL is defined using GSKit (`ssl fips enabled="yes"`) Supply the label which identifies the certificate in the keystore when using SSL authentication.

The default is `IBM TWS 9.5 workstation`, which is the value of the certificate distributed with the product to all customers. This certificate is thus not secure and should be replaced with your own secure certificate. See [Configuring the SSL connection protocol for the network on page 323](#).

ssl encryption cipher = *cipher_class*

Only used if SSL is defined using OpenSSL (`ssl fips enabled="no"`) Define the ciphers that the workstation supports during an SSL connection.

Use one of the common cipher classes listed in [Table 20: Valid encryption cipher classes on page 60](#). The default value is `TLSv1`. No manual intervention is required. If you want to use an OpenSSL cipher class not listed in the table, use the following command to determine if your required class is supported:

```
openssl ciphers class_name
```

where *class_name* is the name of the class you want to use. If the command returns a cipher string, the class can be used.

ssl fips enabled = *yes|no*

Determines whether your entire IBM Workload Scheduler network is enabled for FIPS (Federal Information Processing Standards) compliance. FIPS compliance requires the use of GSKit instead of the default OpenSSL for secure communications. If you enable FIPS (`ssl fips enabled="yes"`) the values for all the SSL attributes that apply to GSKit are automatically applied by IBM Workload Scheduler. If you do not enable FIPS (`ssl fips enabled="no"`), the values for all the SSL attributes that apply to OpenSSL are automatically applied by IBM Workload Scheduler. The default is **no**.



Note: In versions 9.5 and 10.1, FIPS compliance is not complete, because you cannot configure WebSphere Application Server Liberty Base for FIPS compliance. However, you can enable FIPS compliance for your IBM Workload Scheduler static network.

ssl key = *file_name*

Only used if SSL is defined using OpenSSL (`ssl fips enabled="no"`) The name of the private key file.

The default is `TWA_home/TWS/ssl/TWSPrivateKeyFile.pem`. This file is part of the SSL configuration distributed with the product to all customers. It is thus not secure and should be replaced with your own secure SSL configuration. See [Configuring the SSL connection protocol for the network on page 323](#).

ssl key pwd = *file_name*

Only used if SSL is defined using OpenSSL (`ssl fips enabled="no"`) The name of the file containing the password for the stashed key.

The default is `TWA_home/TWS/ssl/TWSPrivateKeyFile.sth`. This file is part of the SSL configuration distributed with the product to all customers. It is thus not secure and should be replaced with your own secure SSL configuration. See [Configuring the SSL connection protocol for the network on page 323](#).

ssl keystore file = *file_name*

Only used if SSL is defined using GSKit (`ssl fips enabled="yes"`). Specify the name of the keystore file used for SSL authentication.

The default is `TWA_home/TWS/ssl/TWSKeyRing.kdb`. This file is part of the SSL configuration distributed with the product to all customers. It is thus not secure and should be replaced with your own secure SSL configuration. See [Configuring the SSL connection protocol for the network on page 323](#).

ssl keystore pwd = *file_name*

Only used if SSL is defined using GSKit (`ssl fips enabled="yes"`). Specify the name of the keystore password file used for SSL authentication.

The default is `TWA_home/TWS/ssl/TWSKeyRing.sth`. This file is part of the SSL configuration distributed with the product to all customers. It is thus not secure and should be replaced with your own secure SSL configuration. See [Configuring the SSL connection protocol for the network on page 323](#).

ssl random seed = *file_name*

Only used if SSL is defined using OpenSSL (`ssl fips enabled="no"`) Specify the pseudo random number file used by OpenSSL on some operating systems. Without this file, SSL authentication might not work correctly.

The default is `TWA_home/TWS/ssl/TWS.rnd`. This file is part of the SSL configuration distributed with the product to all customers. It is thus not secure and should be replaced with your own secure SSL configuration. See [Configuring the SSL connection protocol for the network on page 323](#).

stdlist width = *columns*

Specify the maximum width of the IBM Workload Scheduler console messages. You can specify a column number in the range **1** to **255**. Lines are wrapped at or before the specified column, depending on the presence of imbedded carriage control characters. Specify a negative number or zero to ignore line width. On UNIX® and Linux® operating systems, you should ignore line width if you enable system logging with the **syslog local** option. The default is 0 columns.

switch sym prompt = *prompt*

Specify a prompt for the conman command line after you have selected a different Symphony file with the **setsym** command. The maximum length is 8 characters. The default is **n%**.

sync level = *low|medium|high*

Specify the rate at which IBM Workload Scheduler synchronizes information written to disk. This option affects all mailbox agents and is applicable to UNIX® and Linux® operating systems only. Values can be:

low

Allows the operating system to handle it.

medium

Flushes the updates to disk after a transaction has completed.

high

Flushes the updates to disk every time data is entered.

The default is **low**.

syslog local = *value*

Enables or disables IBM Workload Scheduler system logging for UNIX® and Linux® operating systems only. Specify **-1** to turn off system logging for IBM Workload Scheduler. Specify a number from **0** to **7** to turn on system logging and have IBM Workload Scheduler use the corresponding local facility (LOCAL0 through LOCAL7) for its messages. Specify any other number to turn on system logging and have IBM Workload Scheduler use the USER facility for its messages. The default is -1. See [IBM Workload Scheduler console messages and prompts on page 125](#).

tcp connect timeout = *seconds*

Specify the maximum number of seconds that can be waited to establish a connection through non-blocking socket. The default is 15 seconds.

tcp timeout = *seconds*

Specify the maximum number of seconds that can be waited for the completion of a request on a connected workstation that is not responding. The default is 300 seconds.

this cpu = *workstation_name*

The unique identifier of the workstation. Even when the workstation is subsequently moved to a different folder, the unique identifier remains the same. The name can be a maximum of 16 alphanumeric characters in length and must start with a letter. When a switch is made between the master domain manager and a backup domain manager, using the switchmgr command, the Symphony header value for this cpu is overwritten by the this cpu value in the `localopts` file. The default is the host name of the computer.

timeout = *seconds*

The timeout in seconds to await for the server operation completion was reached. The command continues to run on the server until its completion. The default value is 3600 seconds.

unison network directory = *directory_name*

This parameter applies only to versions of IBM Workload Scheduler prior to version 8.3. Defines the name of the Unison network directory. The default is `TWA_home>../unison/network`.

useropts = *file_name*

If you have multiple instances of IBM Workload Scheduler on a system, use this to identify the `useropts` file that is to be used to store the connection parameters for the instance in which this `localops` file is found. See [Multiple product instances on page 77](#) for more details.

wr enable compression = yes|no

Use this option on fault-tolerant agents. Specify if the fault-tolerant agent can receive the Symphony file in compressed form from the master domain manager. The default is **no**.

wr read = seconds

Specify the number of seconds the Writer process waits for an incoming message before checking for a termination request from Netman. The default is 600 seconds.

wr unlink = seconds

Specify the number of seconds the Writer process waits before exiting if no incoming messages are received. The minimum is 120 seconds. The default is 180 seconds.

Local options file example

The following is an example of a default `localopts` file:



Note: Some parameters might not be present depending upon your version and configuration.

Example

```
#####
# Licensed Materials - Property of IBM* and HCL**
# 5698-WSH
# (C) Copyright IBM Corp. 1998, 2016 All rights reserved.
# (C) Copyright HCL Technologies Ltd. 2016, 2022 All rights reserved.
# * Trademark of International Business Machines
# ** Trademark of HCL Technologies Limited
#####
#
# The IBM Workload Scheduler localopts file defines the attributes of this
# workstation, for various processes.
#
#-----
# General attributes of this workstation:
#
thiscpu=FTA_nc004163
merge stdlists      =yes
stdlist width      =0
syslog local        =-1
restricted stdlists =no
#
#-----
# The attributes of this workstation for the batchman process:
#
bm check file       =120
bm check status     =300
bm look             =15
bm read             =10
bm stats            =off
bm verbose          =off
bm check until      =300
bm check deadline   =30
bm late every       =0
```



```

#
#-----
# The attributes of this workstation for the jobman process:
#
jm job table size =1024
jm look           =300
jm nice           =0
jm promoted nice  =-1   #UNIX
jm promoted priority =AboveNormal #WINDOWS
jm no root        =yes
jm file no root   =no
jm read           =10
#
#-----
# The attributes of this workstation for the TWS mailman process:
#
mm response       =600
mm retrylink      =600
mm sound off      =no
mm unlink         =960
mm cache mailbox  =yes
mm cache size     =512
mm resolve master =no
autostart monman  =yes
mm symphony download timeout =0
#
#-----
# The attributes of this workstation for the netman process:
#
nm mortal         =no
nm port           =35111
nm read           =10
nm retry          =800
#
#-----
# The attributes of this workstation for the writer process:
#
wr read           =600
wr unlink         =180
wr enable compression =no
#
#-----
# Optional attributes of this Workstation for remote database files
#
# mozart directory =           /home/ITAuser/TWA/TWS/mozart
# parameters directory =       /home/ITAuser/TWA
# unison network directory =    /home/ITAuser/TWA/TWS/./unison/network
#
#-----
# The attributes of this workstation for custom formats
#
date format       =1 # The possible values are 0-yyyy/mm/dd, 1-mm/dd/yyyy, 2-dd/mm/yyyy, 3-NLS.
composer prompt   =-
conman prompt     =%
switch sym prompt <=n>%
#
#-----
# The attributes of this workstation for the customization of I/O on mailbox files

```

```

# sync level          =low
#
#-----
# The attributes of this workstation for networking
# tcp timeout        =300 tcp connect timeout=5
#
#-----
# General Secure options
#
SSL auth mode         =caonly
#
# Use "SSL auth string" only if "SSL auth mode" is set to "string"
#
SSL auth string       =twS
#
# The value "yes" for "SSL Fips enabled" forces TWS to use GSKIT, else it uses OpenSSL
# This flag set to "yes" enables the FIPS 140-2 policies. The default value is "no".
#
SSL Fips enabled=yes
#
# Netman full SSL port, use "nm SSL full port" it on if "enSSLFullConnection" is set to "yes"
# the value "0" means port close
#
nm SSL full port=0
#
# Netman SSL port
# the value "0" means port close
#
nm SSL port=33113
#
# End General Secure options
#-----

#-----
# OpenSSL option, TWS uses them if "SSL Fips enabled" is "no" ( the default )
#
SSL key="/home/ITAuser/TWA/TWS/ssl/OpenSSL/TWSClient.key"
SSL certificate="/home/ITAuser/TWA/TWS/ssl/OpenSSL/TWSClient.cer"
SSL key pwd="/home/ITAuser/TWA/TWS/ssl/OpenSSL/password.sth"
SSL CA certificate="/home/ITAuser/TWA/TWS/ssl/OpenSSL/TWSTrustCertificates.cer"
SSL random seed="/home/ITAuser/TWA/TWS/ssl/OpenSSL/TWS.rnd"
SSL Encryption Cipher=TLSv1
CLI SSL cipher=HIGH
#
#CLI SSL server auth =
#CLI SSL server certificate =
#CLI SSL trusted dir =
# End OpenSSL options
#-----

#-----
# GSKIT options, TWS uses them if "SSL Fips enabled" is "yes"
#
SSL keystore file="/home/ITAuser/TWA/TWS/ssl/GSKit/TWSClientKeyStore.kdb"
SSL certificate keystore label="client"
SSL keystore pwd="/home/ITAuser/TWA/TWS/ssl/GSKit/TWSClientKeyStore.sth"
#
#

```

```

CLI SSL keystore file="/home/ITAuser/TWA/TWS/ssl/GSKit/TWSClientKeyStore.kdb"
CLI SSL certificate keystore label="client"
CLI SSL keystore pwd="/home/ITAuser/TWA/TWS/ssl/GSKit/TWSClientKeyStore.sth"
#----- End GSKit options -----

#-----
# The TWS instance has been installed as REMOTE CLI
IS REMOTE CLI = no # yes for a REMOTE CLI installation, no otherwise

#-----
# Attributes for CLI connections
#
# General attributes for CLI connections
#
HOST=nc004113
PROTOCOL=https
PORT=35116
#PROXY          =
#PROXYPORT      =
#TIMEOUT        = 3600      # Timeout in seconds to wait a server response
#CLI SSL SERVER AUTH = yes
FOLLOWLOCATION   = true

#DEFAULTTWS    =
#USEROPTS      =

#-----
# Event Management parameters
#
CAN BE EVENT PROCESSOR = no # yes for MDM and BKM, no otherwise

#-----
# Centralized Agent Update
#
#DownloadDir =

SSL certificate chain      =/home/ITAuser/TWA/TWS/ssl/TWSCertificateChain.crt
merge logtrace            = yes
LOCAL WAS                  = no
mm read                    = 15
tcp connection timeout    = 15
#CLI SSL server auth      =

#CLI SSL server auth      =

#CLI SSL server auth      =
#CLI SSL server auth      =
#CLI SSL server auth      =

#CLI SSL server auth      =
#CLI SSL server auth      =
#CLI SSL server auth      =

#CLI SSL server auth      =
#-----

```

```
# Current Folder
#
#current folder = /
```



Note: The "REMOTE CLI" term indicates the command line client.

Setting user options

Set the user options you require for each user on a workstation who needs them in the `useropts` file. Changes do not take effect until IBM Workload Scheduler is stopped and restarted.

The concept of the `useropts` file is to contain values for `localopts` parameters that must be personalized for an individual user. The files must be located in the `user_home/.TWS` directory of the user. When IBM Workload Scheduler needs to access data from the `localopts` file, it looks first to see if the property it requires is stored only or also in the `useropts` file for the user, always preferring the `useropts` file version of the value of the key. If a property is not specified when invoking the command that requires it, or inside the `useropts` and `localopts` files, an error is displayed.

The main use of the `useropts` file is to store the user-specific connection parameters used to access the command line client (see [Configuring command-line client access authentication on page 117](#)). These are the following keys, which are not stored in the `localopts` file:

username

User name used to access the master domain manager. The user must be defined in the security file on the master domain manager (see [Configuring user authorization \(Security file\) on page 168](#))

password

Password used to access the master domain manager. The presence of the `ENCRYPT` label in the password field indicates that the specified setting has been encrypted; if this label is not present, you must exit and access the interface program again to allow the encryption of that field.

A `useropts` file is created for the `<TWS_user>` during the installation, but you must create a separate file for each user that needs to use user-specific parameters on a workstation.

See [Localopts details on page 55](#) for more detailed information about these options.

Sample useropts file

This is the sample content of a `useropts` file:

```
#
# IBM Workload Scheduler useropts file defines attributes of this Workstation.
#
#-----
# Attributes for CLI connections
USERNAME   = MDMDBE4      # Username used in the connection
PASSWORD   = "ENCRYPT:YEE7cEZs+HE+mEHCsdN0fg==" # Password used in the connection
#HOST      =              # Master hostname used when attempting a connection.
PROTOCOL    = https      # Protocol used to establish a connection with the Master.
#PROTOCOL   = http       # Protocol used to establish a connection with the Master.
```

```

PORT      = 3111      # Protocol port
#PROXY    =
#PROXYPORT =
TIMEOUT   = 120      # Timeout in seconds to wait a server response
#DEFAULTWS =

CLI SSL keystore file      = "$(install_dir)/ssl/MyTWSKeyRing.kdb"
CLI SSL certificate keystore label = "client"
CLI SSL keystore pwd      = "$(install_dir)/ssl/MyTWSKeyRing.sth"
current folder            = /apps

```

The SSL configuration options for the command line client depend on the type of SSL implemented - here GSKit is assumed.



Note: The # symbol is used to comment a line.

Multiple product instances

Because IBM Workload Scheduler supports multiple product instances installed on the same computer, there can be more than one instance of the `useropts` file per user. This is achieved by giving the `useropts` files for a user different names for each instance.

In the `localopts` file of each instance the option named **`useropts`** identifies the file name of the `useropts` file that has to be accessed in the `user_home/.TWS` directory to connect to that installation instance.

This means that, for example, if two IBM Workload Scheduler instances are installed on a computer and the user `operator` is a user of both instances, you could define the `useropts` credentials as follows:

- In the `localopts` file of the *first* instance the local option `useropts = useropts1` identifies the `operator_home/.TWS/useropts1` file containing the connection parameters settings that user `operator` needs to use to connect to the *first* IBM Workload Scheduler instance.
- In the `localopts` file of the *second* IBM Workload Scheduler instance the local option `useropts = useropts2` identifies the `operator_home/.TWS/useropts2` file containing the connection parameters settings that user `operator` needs to use to connect to the *second* IBM Workload Scheduler instance.

Configuring the agent

The configuration settings of the agent are stored in the `JobManager.ini` file.

In a distributed environment, if a gateway is configured to allow the master domain manager or dynamic domain manager to communicate with a dynamic agent located behind a network boundary, then the gateway configuration settings of the agent are contained in the `JobManagerGW.ini` file. This file is almost identical to the `JobManager.ini` file, however, only parameters in the [ITA], [Env], and [ResourceAdvisorAgent] sections require configuration. For these parameters, definitions are given for both the `JobManager.ini` and `JobManagerGW.ini` files.

To find out where these files are located, see the section about installation paths in *IBM Workload Scheduler: Planning and Installation*.

Only a subset of the available parameters is documented, because some parameters are reserved for internal use.

These files are made up of many different sections. Each section name is enclosed between square brackets and each section includes a sequence of `variable = value` statements.

You can customize properties for the following:

- Event-driven workload automation properties
- Log properties
- Trace properties when the agent is stopped. You can also customize traces when the agent is running using the procedure described in [Configuring trace properties when the agent is running on page 84](#).
- Native job executor
- Java™ job executor
- Resource advisor agent
- System scanner

The log messages are written in the following file:

On Windows operating systems:

`<TWA_home>\TWS\stdlist\JM\JobManager_message.log`

On UNIX and Linux operating systems:

`<TWA_DATA_DIR>/stdlist/JM/JobManager_message.log`

The trace messages are written in the following file:

On Windows operating systems:

- `<TWA_home>\TWS\stdlist\JM\ITA_trace.log`
- `<TWA_home>\TWS\stdlist\JM\JobManager_trace.log`
- `<TWA_home>\TWS\JavaExt\logs\javaExecutor0.log`

On UNIX and Linux operating systems:

- `<TWA_DATA_DIR>/stdlist/JM/ITA_trace.log`
- `<TWA_DATA_DIR>/stdlist/JM/JobManager_trace.log`
- `<TWA_DATA_DIR>/JavaExt/logs/javaExecutor0.log`

Logging information about job types with advanced options

You can use the `logging.properties` file to configure the logging process for job types with advanced options, with the exception of the Executable and Access Method job types.

The `logging.properties` file is located on the IBM® Z Workload Scheduler Agent, located in the following path:

On Windows operating systems:

`<TWA_home>/TWS/JavaExt/cfg/logging.properties`

On UNIX and Linux operating systems:

```
<TWA_DATA_DIR>/JavaExt/cfg/logging.properties
```

After installation, this file is as follows:

```
# Specify the handlers to create in the root logger
# (all loggers are children of the root logger)
# The following creates two handlers
handlers = java.util.logging.ConsoleHandler,
           java.util.logging.FileHandler

# Set the default logging level for the root logger
.level = INFO

# Set the default logging level for new ConsoleHandler instances
java.util.logging.ConsoleHandler.level = INFO

# Set the default logging level for new FileHandler instances
java.util.logging.FileHandler.level
= ALL
java.util.logging.FileHandler.pattern
= C:\TWA_home\TWS\JavaExt\logs\javaExecutor%g.log
java.util.logging.FileHandler.limit
= 1000000
java.util.logging.FileHandler.count
= 10

# Set the default formatter for new ConsoleHandler instances
java.util.logging.ConsoleHandler.formatter =
    java.util.logging.SimpleFormatter
java.util.logging.FileHandler.formatter =
    java.util.logging.SimpleFormatter

# Set the default logging level for the logger named com.mycompany
com.ibm.scheduling = INFO
```

You can customize:

- The logging level (from INFO to WARNING, ERROR, or ALL) in the following keywords:

```
.level
```

Defines the logging level for the internal logger.

```
com.ibm.scheduling
```

Defines the logging level for the job types with advanced options. To log information about job types with advanced options, set this keyword to ALL.

- The path where the logs are written, specified by the following keyword:

```
java.util.logging.FileHandler.pattern
```

Not all the properties in the `JobManager.ini` and `JobManagerGW.ini` files can be customized. For a list of the configurable properties, see the following sections:

- [Configuring log message properties \[JobManager.Logging.cilog\] on page 81.](#)
- [Configuring trace properties when the agent is stopped \[JobManager.Logging.cilog\] on page 82.](#)
- [Configuring common launchers properties \[Launchers\] on page 87.](#)
- [Configuring properties of the native job launcher \[NativeJobLauncher\] on page 89.](#)
- [Configuring properties of the Java job launcher \[JavaJobLauncher\] on page 92.](#)
- [Configuring properties of the Resource advisor agent \[ResourceAdvisorAgent\] on page 92.](#)
- [Configuring properties of the System scanner \[SystemScanner\] on page 95](#)
- [Configuring environment variables \[Env\] on page 95](#)
- the section about configuring properties of event-driven workload automation [EventDrivenWorkload] in *IBM Z Workload Scheduler: Scheduling End-to-end with z-centric Capabilities*

Configuring general properties [ITA]

About this task

In the `JobManager.ini` or `JobManagerGW.ini` file, you can add some general properties to the following section:

```
[ITA]
```

You can add or modify the following properties:

ActionPollers

The number of the thread processes started on the gateway workstation to communicate with the broker server installed on the master domain manager or dynamic domain manager. The default value is 1. Specify this value if you have more than 100 dynamic agents that communicate with the broker server installed on the master domain manager or dynamic domain manager by using the same gateway. Restart the agent after the property change.

http_proxy

The URL of the proxy configured in a distributed environment through which agents or gateways communicate to the broker server installed on the master domain manager or dynamic domain manager. The value is

`https_proxy =http://<proxy_workstation>:<proxy_workstation_port>`, where:

- `<proxy_workstation>` is the fully qualified host name of the workstation where the proxy is configured.
- `<proxy_workstation_port>` is the port number of the workstation where the proxy is configured.

Restart the agent after the property change.

DebugDir

You can use this parameter to enable tracing for the dynamic agent to help determine what information is being sent to and from a dynamic agent workstation. Perform the following steps:

1. Create a directory to dump the files sent and received by the dynamic agent, for example `/tmp/DA_DD`. This directory needs to be writable by the user owning the dynamic agent.
2. Add the **DebugDir** parameter to the path of the directory you created, for example:

```
DebugDir = /tmp/DA_DD
```

3. Restart the dynamic agent using the following commands:

```
ShutDownLwa
StartUpLwa
```

Remember to monitor the debug directory on a regular basis to ensure it does not become too large. No automatic check is performed on the debug director. For more information, see [Enable packet tracing for dynamic agent using ITA parameter DebugDir](#).

Configuring log message properties [JobManager.Logging.cclog]

About this task

To configure the logs, edit the [JobManager.Logging.cclog] section in the `JobManager.ini` file. This procedure requires that you stop and restart the IBM Workload Scheduler agent

The section containing the log properties is named:

```
[JobManager.Logging.cclog]
```

You can change the following properties:

JobManager.loggerhd.fileName

The name of the file where messages are to be logged. the default value is

On Windows operating systems

`POSIXHOME\stdlist\JM\JOBMANAGER-FFDC` where *POSIXHOME* is the installation directory.

On UNIX operating systems

`$(TWA_DATA_DIR)/stdlist/JM/JobManager_message.log`

JobManager.loggerhd.maxFileBytes

The maximum size that the log file can reach. The default is **1024000** bytes.

JobManager.loggerhd.maxFiles

The maximum number of log files that can be stored. The default is **3**.

JobManager.loggerhd.fileEncoding

By default, log files for the agent are coded in UTF-8 format. If you want to produce the log in a different format, add this property and specify the required codepage.

JobManager.loggerfl.level

The amount of information to be provided in the logs. The value ranges from 3000 to 7000. Smaller numbers correspond to more detailed logs. The default is **3000**.

JobManager.ffdc.maxDiskSpace

Exceeding this maximum disk space, log files collected by the first failure data capture mechanism are removed, beginning with the oldest files first.

JobManager.ffdc.baseDir

The directory to which log and trace files collected by the ffdc tool are copied. The default directory is

On Windows operating systems

`POSIXHOME\stdlist\JM\JobManager_message.log` where `POSIXHOME` is the installation directory.

On UNIX operating systems

`$(TWA_DATA_DIR)/stdlist/JM/JobManager_message.log`

JobManager.ffdc.filesToCopy

Log and trace files (`JobManager_message.log` and `JobManager_trace.log`) collected by the ffdc tool located in `<TWA_home>\TWS\stdlist\JM`. For example, `JobManager.ffdc.filesToCopy = "/opt/IBM/TWA_<TWS_user>/TWS/stdlist/JM/JobManager_message.log" "/opt/IBM/TWA_<TWS_user>/TWS/stdlist/JM/JobManager_trace.log"`

When a message is logged (`JobManager.ffdc.triggerFilter = JobManager.msgIdFilter`) that has an ID that matches the pattern "AWSITA*E" (`JobManager.msgIdFilter.msgIds = AWSITA*E`), which corresponds to all error messages, then the log and trace files (`JobManager.ffdc.filesToCopy = "/opt/IBM/TWA_<TWS_user>/TWS/stdlist/JM/JobManager_message.log" "/opt/IBM/TWA_<TWS_user>/TWS/stdlist/JM/JobManager_trace.log"`) are copied (`JobManager.ffdc.className = ccg_ffdc_filecopy_handler`) to the directory `JOBMANAGER-FFDC` (`JobManager.ffdc.baseDir = /opt/IBM/TWA_<TWS_user>/TWS/stdlist/JM/JOBMANAGER-FFDC`). If the files copied exceed 10 MB (`JobManager.ffdc.maxDiskSpace = 10000000`), then the oldest files are removed first (`JobManager.ffdc.quotaPolicy = QUOTA_AUTODELETE`).

After installing the z-centric agent or dynamic agent on Windows 2012, the `JobManager_message.log` might not be created. In this case, perform the following procedure:

1. Stop the agent.
2. Create a backup copy of `JobManager.ini`, and edit the original file by changing the row:

```
JobManager.loggerhd.className = ccg_multiproc_filehandler
```

to

```
JobManager.loggerhd.className = ccg_filehandler
```

3. Restart the agent.

Configuring trace properties when the agent is stopped [JobManager.Logging.cilog]

How to configure the trace properties when the agent is stopped.

To configure the trace properties when the agent is stopped, edit the [JobManager.Logging] section in the `JobManager.ini` file and then restart the IBM Workload Scheduler agent.

The section containing the trace properties is named:

```
[JobManager.Logging.cclog]
```

You can change the following properties:

JobManager.trhd.fileName

The name of the trace file.

JobManager.trhd.maxFileBytes

The maximum size that the trace file can reach. The default is 1024000 bytes.

JobManager.trhd.maxFiles

The maximum number of trace files that can be stored. The default is 3.

JobManager.trfl.level

Determines the type of trace messages that are logged. Change this value to trace more or fewer events, as appropriate, or on request from IBM Software Support. Valid values are:

DEBUG_MAX

Maximum tracing. Every trace message in the code is written to the trace logs.

INFO

All *informational*, *warning*, *error* and *critical* trace messages are written to the trace. The default value.

WARNING

All *warning*, *error* and *critical* trace messages are written to the trace.

ERROR

All *error* and *critical* trace messages are written to the trace.

CRITICAL

Only messages which cause the agent to stop are written to the trace.

The output trace (`JobManager_trace.log`) is provided in XML format.

After installing the z-centric agent or dynamic agent on Windows 2012, the `JobManager_trace.log` might not be created. In this case, perform the following procedure:

1. Stop the agent.
2. Create a backup copy of `JobManager.ini`, and edit the original file by changing the row:

```
JobManager.trhd.className = ccg_multiproc_filehandler
```

to

```
JobManager.trhd.className = ccg_filehandler
```

3. Restart the agent.

Configuring trace properties when the agent is running

Use the **twstrace** command to set the trace on the agent when it is running.

Using the **twstrace** command, you can perform the following actions on the agent when it is running:

- [See command usage and verify version on page 84.](#)
- [Enable or disable trace on page 84.](#)
- Set the traces to a specific level, specify the number of trace files you want to create, and the maximum size of each trace file. See [Set trace information on page 85.](#)
- [Show trace information on page 85.](#)
- Collect trace files, message files, and configuration files in a compressed file using the command line. See [Collect trace information on page 86.](#)
- Collect trace files, message files, and configuration files in a compressed file using the Dynamic Workload Console. See [Retrieving IBM Workload Scheduler agent traces from the Dynamic Workload Console.](#)

You can also configure the traces when the agent is not running by editing the [JobManager.Logging] section in the `JobManager.ini` file as described in [Configuring the agent](#) section. This procedure requires that you stop and restart the agent.

twstrace command

Use the **twstrace** command to configure traces, and collect logs, traces, and configuration files (ita.ini and jobManager.ini) for agents. You collect all the information in a compressed file when it is running without stopping and restarting it.

See command usage and verify version

To see the command usage and options, use the following syntax.

Syntax

```
twstrace -u | -v
```

Parameters

-u

Shows the command usage.

-v

Shows the command version.

Enable or disable trace

To set the trace to the maximum or minimum level, use the following syntax.

Syntax

twstrace -enable | -disable

Parameters**-enable**

Sets the trace to the maximum level. The maximum level is **1000**.

-disable

Sets the trace to the minimum level. The minimum level is **3000**.

Set trace information

To set the trace to a specific level, specify the number of trace files you want to create, and the maximum size the trace files can reach, use the following syntax.

Syntax

twstrace [-level <level_number>] [-maxFiles <files_number>] [-maxFileBytes <bytes_number>]

Parameters**-level <level_number>**

Sets the trace level. Specify a value in the range from 1000 to 3000, which is also the default value. Note that if you set this parameter to 3000, you have the lowest verbosity level and the fewest trace messages. To have a better trace level, with the most verbose trace messages and the maximum trace level, set it to **1000**.

-maxFiles <files_number>

Specify the number of trace files you want to create.

-maxFileBytes <bytes_number>

Set the maximum size in bytes that the trace files can reach. The default is **1024000** bytes.

Show trace information

To display the current trace level, the number of trace files, and the maximum size the trace files can reach, use the following syntax.

Syntax

twstrace -level | -maxFiles | -maxFileBytes

Parameters**-level**

See the trace level you set.

-maxFiles

See the number of trace files you create.

-maxFileBytes

See the maximum size you set for each trace file

Example**Sample**

The example shows the information you receive when you run the following command:

```
twstrace -level -maxFiles -maxFileBytes
AWSITA176I The trace properties are: level="1000",
max files="3", file size="1024000".
```

Collect trace information

To collect the trace files, the message files, and the configuration files in a compressed file, use the following syntax.

Syntax

```
twstrace -getLogs [ -zipFile <compressed_file_name> ] [ -host <host_name> ] [ -protocol {http | https} [ -port <port_number> ] [ -iniFile <ini_file_name> ]
```

Parameters**-zipFile <compressed_file_name>**

Specify the name of the compressed file that contains all the information, that is logs, traces, and configuration files (ita.ini and jobManager.ini) for the agent. The default is **logs.zip**.

-host <host_name>

Specify the host name or the IP address of the agent for which you want to collect the trace. The default is **localhost**.

-protocol http|https

Specify the protocol of the agent for which you are collecting the trace. The default is the protocol specified in the **.ini** file of the agent.

-port <port_number>

Specify the port of the agent. The default is the port number of the agent where you are running the command line.

-iniFile <ini_file_name>

Specify the name of the **.ini** file that contains the SSL configuration of the agent for which you want to collect the traces. If you are collecting the traces for a remote agent for which you customized the security certificates, you must import the certificate on the local agent and specify the name of the **.ini** file that contains this configuration. To do this, perform the following actions:

1. Extract the certificate from the keystore of the remote agent.
2. Import the certificate in a local agent keystore. You can create an ad hoc keystore whose name must be **TWSClientKeyStore.kdb**.

3. Create an **.ini** file in which you specify:

- **0** in the **tcp_port** property as follows:

```
tcp_port=0
```

- The port of the remote agent in the **ssl_port** property as follows:

```
ssl_port=<ssl_port>
```

- The path to the keystore you created in Step 2 on page 86 in the **key_repository_path** property as follows:

```
key_repository_path=<local_agent_keystore_path>
```

Configuring common launchers properties [Launchers]

About this task

In the `JobManager.ini` file, the section containing the properties common to the different launchers (or executors) is named:

```
[Launchers]
```

The following properties are available:

BaseDir

The installation path of the IBM Workload Scheduler agent. Do not modify this value.

CommandHandlerMinThreads

Indicates the maximum number of commands that can be run on the agent concurrently. Limits to the number of jobs vary depending on the resources of your workstation, however consider that operations on comdhandler are usually short. The default is **20**. Usually, there is no need to modify this setting, even if you plan a very high workload on the agent. You might want to change it if many commands are run concurrently on the agent, for example, many concurrent requests to retrieve job logs.

CommandHandlerMaxThreads

Indicates the maximum number of commands that can be run on the agent concurrently. Limits to the number of jobs vary depending on the resources of your workstation, however consider that operations on comdhandler are usually short. The default is **100**. Usually, there is no need to modify this setting, even if you plan a very high workload on the agent. You might want to change it if many commands are run concurrently on the agent, for example, many concurrent requests to retrieve job logs.

CpaHeartBeatTimeSeconds

The polling interval in seconds used to verify if the **agent** process is still up and running. If the agent process is inactive the product stops also the **JobManager** process. The default is **30**. Modify only if you use dynamic pools with CPU-based requirements or optimization policies. With a lower value, the agent reacts quickly to CPU modifications, but this might cause unstable values in case of CPU spikes. Lower values causes a higher use of resources on the agent.

DirectoryPermissions

The access rights assigned to the agent for creating directories when running jobs. The default is **0755**. Supported values are UNIX-format entries in hexadecimal notation.

DownloadDir

The name of the directory where the fix pack installation package or upgrade image for fault-tolerant agents or dynamic agents is downloaded during the centralized agent update process. If not specified, the following default directory is used:

On Windows operating systems:

```
<TWA_home>\TWS\stdlist\JM\download
```

On UNIX operating systems:

```
<TWA_home>/TWS/stdlist/JM/download
```

The centralized agent update process does not apply to z-centric agents.

ExecutorsMaxThreads

Specifies the maximum number of jobs the dynamic agent can run concurrently. For example, to allow the dynamic agent to run a maximum of 500 jobs concurrently, set this parameter to **500**. The default is **400**.

ExecutorsMinThreads

Specifies the minimum number of jobs the dynamic agent can run concurrently. For example, to allow the dynamic agent to run a minimum of 500 jobs concurrently, set this parameter to **500**. The default is **38**. Modify if the number of expected concurrent jobs is much higher than 38. The agent dynamically allocates more threads if necessary, until it reaches the value specified in **ExecutorsMaxThreads**.

FilePermissions

The access rights assigned to the agent for creating files when running jobs. The default is **0755**. Supported values are UNIX-format entries in hexadecimal notation.

MaxAge

The number of days that job logs are kept (in path *TWA_home/TWS/stdlist/JM*) before being deleted. The default is **30**. Possible values range from a minimum of 1 day.

NotifierMaxThreads

Notifier threads are in charge of notifying the dynamic workload broker of each status change in a job. This parameter specifies the maximum number of job status changes that can be notified to the dynamic workload broker.

NotifierMinThreads

Notifier threads are in charge of notifying the dynamic workload broker of each status change in a job. This parameter specifies the minimum number of job status changes that can be notified to the dynamic workload broker. The default value is **3**. Modify this parameters only in case of unexpected errors and after consulting with software support team.

SpoolDir

The path to the folder containing the jobstore and outputs. The default is:

```
value of BaseDir/stdlist/JM
```

StackSizeBytes

The size of the operating system stack in bytes. The default is **DEFAULT**, meaning that the **agent** uses the default value for the operating system. Do not modify this parameter unless instructed to do so by the software support team. Incorrect values can cause the agent to crash.

Configuring properties of the native job launcher [NativeJobLauncher]

About this task

In the `JobManager.ini` file, the section containing the properties of the native job launcher is named:

```
[NativeJobLauncher]
```

You can change the following properties:

AllowRoot

Applies to UNIX™ systems only. Specifies if the root user can run jobs on the agent. It can be `true` or `false`. The default is `false`. This property does not apply to IBM i, use the `AllowQSECOFR` option instead

AllowQSECOFR

Applies to IBM i systems only. Specifies if QSECOFR user can run jobs on the agent. It can be `true` or `false`. The default is `true`. Add a line like `AllowQSECOFR = false` to the `JobManager.ini` file to deny job execution to QSECOFR.

CheckExec

If `true`, before launching the job, the agent checks both the availability and the execution rights of the binary file. The default is `true`.

DefaultWorkingDir

Specifies the working directory of native jobs. You can also specify the value for the working directory when creating or editing the job definition in the Workload Designer. When specified in the Workload Designer, this value overrides the value specified for the **DefaultWorkingDir** property. If you do not specify any working directories, the `<TWS_home>\bin` directory is used.

JobUnspecifiedInteractive

Applies to Windows™ operating systems only. Specifies if native jobs are to be launched in interactive mode. It can be `true` or `false`. The default is `false`.

KeepCommandTraces

Set to `true` to store the traces of the method invocation for actions performed on a job definition, for example, when selecting from a picklist. These files are stored in the path `/opt/IBM/TWA_<TWS_user>/TWS/stdlist/JM/r3batch_cmd_exec`. The default setting is `false`.

KeepJobCommandTraces

Set to `true` to store the traces of the method invocation for actions performed on a job instance, for example, viewing a spool list. These files are stored in the .zip file of the job instance. The default setting is `true`.

LoadProfile

Applies to agents on Windows servers only. Specifies if the user profile is to be loaded. It can be `true` or `false`. The default is `true`.

MonitorQueueName

Specifies the name of the queue where the IBM i jobs are monitored. If you do not specify this property, the default queue (QBATCH) is used.

PortMax

The maximum range of the port numbers used by the task launcher to communicate with the Job Manager. The default is 0, meaning that the operating system assigns the port automatically.

PortMin

The minimum range of the port numbers used by the task launcher to communicate with the Job Manager. The default is 0, meaning that the operating system assigns the port automatically.

PostJobExecScriptPathName

The fully qualified path of the script file that you want to run when the job completes. By default, this property is not present in the `JobManager.ini` file. If you do not specify any file path or the script file doesn't exist, no action is taken.

This property applies to dynamic agent and z/OS agent. For details about running a script when a job completes, see *User's Guide and Reference*.

PromotedNice

Used in workload service assurance. This property is not supported on the Agent for z/OS.

For UNIX and Linux operating systems only, assigns the priority value to a critical job that needs to be promoted so that the operating system processes it before others. Applies to critical jobs or predecessors that need to be promoted so that they can start at their critical start time.

Boundary values vary depending upon each specific platform, but generally lower values correspond to higher priority levels and vice versa. The default is -1.

Be aware that:

- The promotion process is effective with negative values only. If you set a positive value, the system runs it with the -1 default value.
- An out of range value (for example -200), prompts the operating system to automatically promote the jobs with the lowest allowed nice value.
- Overusing the promotion mechanism (that is, defining an exceedingly high number of jobs as mission critical and setting the highest priority value here) might overload the operating system, negatively impacting the overall performance of the workstation.

PromotedPriority

Used in workload service assurance. This property is not supported on the Agent for z/OS.

For Windows operating systems only, sets to this value the priority by which the operating system processes a critical job when it is promoted. Applies to critical jobs or predecessors that need to be promoted so that they can start at their critical start time. Valid values are:

- High
- AboveNormal (the default)
- Normal
- BelowNormal
- Low Or Idle

Note that if you set a lower priority value than the one non-critical jobs might be assigned, no warning is given.

RequireUserName

When `true`, requires that you add the user name in the JSDL job definition.

When `false`, runs with the user name used by job manager, that is:

- `TWS_user` on UNIX™ and Linux™ systems
- The local system account on Windows™ systems

The default is `false`.

RunExecutablesAsIBMiJobs

If you set this property to `true`, you can define IBM i jobs as generic jobs without using the XML definition. Generic jobs are automatically converted to IBM i jobs. As a side effect, generic jobs cannot be run when this parameter is enabled (`RunExecutablesAsIBMiJobs=true`). There is no default value because this property is not listed in the `JobManager.ini` file after the agent installation.

If you set this property to `true`, ensure that the user you used to install the agent has been granted the `*ALLOBJ` special authority.

ScriptSuffix

The suffix to be used when creating the script files. It is:

```
.cmd
```

For Windows™

```
.sh
```

For UNIX™

VerboseTracing

Enables verbose tracing. It is set to `true` by default.

Configuring properties of the Java™ job launcher [JavaJobLauncher]

About this task

In the `JobManager.ini` file, the section containing the properties of the Java™ job launcher is named:

```
[JavaJobLauncher]
```

You can change the following properties:

JVMDir

The path to the virtual machine used to start job types with advanced options. You can change the path to another compatible Java™ virtual machine.

JVMOptions

The options to provide to the Java™ Virtual Machine used to start job types with advanced options. Supported keywords for establishing a secure connection are:

- `https.proxyHost`
- `https.proxyPort`

Supported keywords for establishing a non-secure connection are:

- `Dhttp.proxyHost`
- `Dhttp.proxyPort`

For example, to set job types with advanced options, based on the default JVM http protocol handler, to the unauthenticated proxy server called with name `myproxyserver.mycompany.com`, define the following option:

```
JVMOptions = -Dhttp.proxyHost=myproxyserver.mycompany.com -Dhttp.proxyPort=80
```

Configuring properties of the Resource advisor agent [ResourceAdvisorAgent]

About this task

In the `JobManager.ini` and `JobManagerGW.ini` files, the section containing the properties of the Resource advisor agent is named:

```
[ResourceAdvisorAgent]
```

You can change the following properties:

BackupResourceAdvisorUrls

The list of URLs returned by the IBM Workload Scheduler master in a distributed environment or by the dynamic domain manager either in a z/OS or in a distributed environment. The agent uses this list to connect to the master or dynamic domain manager.

CPUScannerPeriodSeconds

The time interval that the Resource advisor agent collects resource information about the local CPU. The default value is every 10 seconds.

FullyQualifiedHostname

The fully qualified host name of the agent. It is configured automatically at installation time and is used to connect with the master in a distributed environment or with the dynamic domain manager in a z/OS or in a distributed environment. Edit only if the host name is changed after installation.

NotifyToResourceAdvisorPeriodSeconds

The time interval that the Resource advisor agent forwards the collected resource information to the Resource advisor. The default value is every 119 seconds.

ResourceAdvisorUrl

JobManager.ini

The URL of the master in a distributed environment, or of the dynamic domain manager in a z/OS or in a distributed environment, that is hosting the agent. This URL is used until the server replies with the list of its URLs. The value is `https://$(tdwb_server):$(tdwb_port)/JobManagerRESTWeb/JobScheduler/resource`, where:

\$(tdwb_server)

is the fully qualified host name of the master in a distributed environment or of the dynamic domain manager either in a z/OS or in a distributed environment.

\$(tdwb_port)

is the port number of the master in a distributed environment or of the dynamic domain manager either in a z/OS or in a distributed environment.

It is configured automatically at installation time. Edit only if the host name or the port number are changed after installation, or if you do not use secure connection (set to `http`). If you set the port number to zero, the resource advisor agent does not start. The port is set to zero if at installation time you specify that you will not be using the master in a distributed environment or the dynamic domain manager either in a z/OS or in a distributed environment.

In a distributed environment, if **-gateway** is set to either `local` or `remote`, then this is the URL of the dynamic agent workstation where the gateway resides and through which the dynamic

agents communicate. The value is `https://$(tdwb_server):$(tdwb_port)/ita/JobManagerGW/JobManagerRESTWeb/JobScheduler/resource`, where:

`$(tdwb_server)`

The fully qualified host name of the dynamic agent workstation where the gateway resides and through which the dynamic agent communicates with the dynamic workload broker.

`$(tdwb_port)`

The port number of the dynamic agent workstation where the gateway resides and through which the dynamic agent communicates with the dynamic workload broker.

JobManagerGW.ini

In a distributed environment, if **-gateway** is set to `local`, then **ResourceAdvisorUrl** is the URL of the master or dynamic domain manager. The value is `https://$(tdwb_server):$(tdwb_port)/JobManagerRESTWeb/JobScheduler/resource`, where:

`$(tdwb_server)`

The fully qualified host name of the master or dynamic domain manager.

`$(tdwb_port)`

The port number of the master or dynamic domain manager.

ScannerPeriodSeconds

The time interval that the Resource advisor agent collects information about all the resources in the local system other than CPU resources. The default value is every 120 seconds.

The resource advisor agent, intermittently scans the resources of the machine (computer system, operating system, file systems and networks) and periodically sends an update of their status to the master or dynamic domain manager either in a z/OS or in a distributed environment.

The CPU is scanned every `CPUScannerPeriodSeconds` seconds, while all the other resources are scanned every `ScannerPeriodSeconds` seconds. As soon as one of the scans shows a significant change in the status of a resource, the resources are synchronized with the master in a distributed environment or the dynamic domain manager either in a z/OS or in a distributed environment. The following is the policy followed by the agent to tell if a resource attribute has significantly changed:

- A resource is added or deleted
- A string attribute changes its value
- A CPU value changes by more than `DeltaForCPU`
- A file system value changes by more than `DeltaForDiskMB` megabytes
- A Memory value changes by more than `DeltaForMemoryMB` megabytes

If there are no significant changes, the resources are synchronized with the IBM Workload Scheduler master in a distributed environment or with the dynamic domain manager either in a z/OS or in a distributed environment every `NotifyToResourceAdvisorPeriodSeconds` seconds.

Configuring properties of the System scanner [SystemScanner]

About this task

In the `JobManager.ini` file, the section containing the properties of the System scanner is named:

```
[SystemScanner]
```

You can change the following properties:

CPUSamples

The number of samples used to calculate the average CPU usage. The default value is 3.

DeltaForCPU

The change in CPU usage considered to be significant when it becomes higher than this percentage (for example, DeltaForCPU is 20 if the CPU usage changes from 10 percent to 30 percent). The default value is 20 percent.

DeltaForDiskMB

The change in use of all file system resources that is considered significant when it becomes higher than this value. The default value is 100 MB.

DeltaForMemoryMB

The change in use of all system memory that is considered significant when it becomes higher than this value. The default value is 100 MB.

Configuring environment variables [Env]

About this task

Add the section `[Env]` to the `JobManagerGW.ini` configuration file and insert the environment variables that you need in your dynamic scheduling environment.

Regular maintenance

Regular maintenance refers to the mechanisms that are used on your dynamic workstation agents to free up storage space and improve performance.

Unlike fault-tolerant agents where maintenance tasks must be performed manually using the `rmstdlist` utility command, you can have regular maintenance performed on your dynamic agent workstations to keep disk space under control by configuring the following parameters as appropriate.

Table 21. Agent configuration parameters

File	Parameter	Description
JobManager.ini located in the path	MaxAge	The number of days that job logs are kept before being deleted. The default is 2. Possible values range from a minimum of 1 day.
On UNIX™ operating systems		
<i>TWA_DATA_DIR</i> /IT TA/cpa/config/JobManager.ini	JobManager.log gerhd.maxFileBytes	The maximum size that the log file can reach. The default is 1024000 bytes.
On Windows™ operating systems		
<i>TWA_home</i> \TWS\IT TA\cpa\config\JobManager.ini	JobManager.log gerhd.maxFiles	The maximum number of log files that can be stored in the <i>stdlist/JM</i> directory. The default is 3.
	JobManager.ffdc .maxDiskSpace	The maximum disk space reached, by the log files collected by the First Failure Data Capture tool, after which the oldest files are removed.
	JobManager.trhd .maxFileBytes	The maximum size that the log file can reach. The default is 10240000 bytes.
	JobManager.trhd .maxFiles	The maximum number of log files that can be stored. The default is 5.
logging.properties located in the path	java.util.logging. FileHandler.limit	The maximum amount to write log messages to a file. Default value is 1000000 (bytes)
On UNIX™ operating systems		
<i>TWA_DATA_DIR</i> /T WS/JavaExt/cfg/	java.util.logging. FileHandler.count	The number of output files to cycle through. Default value is 10.
On Windows™ operating systems		
<i>TWA_home</i> \TWS\Java Ext\cfg		
Logs related to jobs with advanced options.		

Configuring the dynamic workload broker server on the master domain manager and dynamic domain manager

About this task

You can perform these configuration tasks after completing the installation of your master domain manager, dynamic domain manager, and dynamic agents, and any time that you want to change or tune specific parameters in your environment.

The configuration parameters for the dynamic workload broker server are defined by default at installation time. You modify a subset of these parameters using the files that are created when you install dynamic workload broker. The following files are created in the path:

On Windows systems

<TWA_home>\broker\config

On UNIX systems

<TWA_DATA_DIR>/broker/config

ResourceAdvisorConfig.properties

Contains configuration information about the **Resource Advisor**. For more information, see [ResourceAdvisorConfig.properties file on page 99](#).

JobDispatcherConfig.properties

Contains configuration information about the **Job Dispatcher**. For more information, see [JobDispatcherConfig.properties file on page 101](#).

BrokerWorkstation.properties

Contains configuration information about the broker server. [BrokerWorkstation.properties file on page 104](#)

CLConfig.properties

Contains configuration information for the dynamic workload broker command line. This file is described in the section about Command-line configuration file in *User's Guide and Reference*.

audit.properties

Contains options for configuring the auditing of events. This file is documented in the the section about dynamic workload scheduling audit in *Troubleshooting Guide*.

You can modify a subset of the parameters in these files to change the following settings:

- Heartbeat signal from the agents.
- Time interval for job allocation to resources
- Time interval for notifications on resources
- Polling time when checking the status of remote engine workstations
- Maximum number of results when allocating jobs to global resources
- Encryption of any passwords sent in the JSDL definitions
- Time interval for retrying the operation after a **Job Dispatcher** failure
- Time interval for retrying the operation after a client notification failure
- Archive settings for job data

- Job history settings
- Command line properties (see *IBM Workload Scheduler: Scheduling Workload Dynamically*)

The editable parameters are listed in the following sections. If you edit any parameters that are not listed, the product might not work. After modifying the files, you must stop and restart WebSphere Application Server Liberty Base.

Maintaining the dynamic workload broker server on the master domain manager and dynamic domain manager

About this task

Because one dynamic workload broker server is installed with your master domain manager and dynamic domain manager, and one server with every backup manager, you have at least two servers present in your IBM Workload Scheduler network. The server running with the master domain manager is the only one active at any time. The servers installed in the backup managers are idle until you switch managers, and the server in the new manager becomes the active server. To have a smooth transition from one server to another, when you switch managers, it is important that you keep the same configuration settings in the `ResourceAdvisorConfig.properties` and `JobDispatcherConfig.properties` files in all your servers. When you make a change in any of these files of your running dynamic workload broker server, remember to apply the same change also in the dynamic workload broker server idling on your backup manager.

Some of the settings for the dynamic workload broker server are stored in the local **BrokerWorkstation.properties** file and also in the IBM Workload Scheduler database. When you switch to the backup master domain manager or dynamic domain manager, the dynamic workload broker server settings are automatically updated on the backup workstation. For more information about the **BrokerWorkstation.properties** file, see [BrokerWorkstation.properties file on page 104](#).



Note: The database is automatically populated with the information from the active workstation, regardless of whether it is the manager or the backup workstation. For example, if you modify the dynamic workload broker server settings on the backup master domain manager or dynamic domain manager, this change is recorded in the database. When you switch back to the manager workstation, the change is applied to the master domain manager or dynamic domain manager and the related local settings are overwritten.

It is important that you also keep the data pertinent to every dynamic workload broker server up-to-date. If you change the host name or port number of any of your dynamic workload broker servers, use the `exportserverdata` and `importserverdata` commands from the dynamic workload broker command line to record these changes in the IBM Workload Scheduler database. For information about these commands, see *Scheduling Workload Dynamically*.

The database records for your workload broker workstations all have LOCALHOST as the host name of the workstation. Leave the record as-is. Do not replace LOCALHOST with the actual host name or IP address of the workstation. LOCALHOST is used intentionally to ensure that the jobs submitted from IBM Workload Scheduler are successfully sent to the new local dynamic workload broker when you switch the master domain manager or dynamic domain manager.

Enabling unsecure communication with the dynamic workload broker server

About this task

By default, the dynamic workload broker server uses secure communication. You might need to enable unsecure communication, even though this type of communication is not recommended.

To enable unsecure communication with the dynamic workload broker server, perform the following steps on the master domain manager:

1. Run the `exportserverdata` command located in `TWA_home/TDWB/bin`:

```
exportserverdata -dbUsr db_instance_admin -dbPwd db_instance_admin_pwd
```

2. Open the resulting `server.properties` file in a flat-text editor.
3. Copy the following line:

```
https://hostname:port/JobManagerRESTWeb/JobScheduler
```

4. Change the copied line by replacing **https** with **http**:

```
http://hostname:port/JobManagerRESTWeb/JobScheduler
```

The file now contains two lines specifying the connection mode, one line specifying the https mode and one line specifying the http mode.

5. Save the file.
6. Import the new data with the `importserverdata` command located in `TWA_home/TDWB/bin`:

```
importserverdata -dbUsr db_instance_admin -dbPwd db_instance_admin_pwd
```

For more information about the `exportserverdata` and `importserverdata` commands, see *IBM Workload Scheduler: Scheduling Workload Dynamically*.

ResourceAdvisorConfig.properties file

The parameters in this file affect the following dynamic workload broker server settings:

- Heartbeat signal from the agents
- Time interval for job allocation to resources
- Time interval for notifications on resources
- Polling time when checking the status of remote engine workstations
- Maximum number of results when allocating jobs to global resources

You can modify the following parameters in the `ResourceAdvisorConfig.properties` file:

DatabaseCheckInterval

Specifies the time interval within which the dynamic workload broker server checks the availability of the database. The default value is **180** seconds.

ResourceAdvisorURL

Specifies the URL of the **Resource Advisor**.

RaaHeartBeatInterval

Specifies the time interval within which the **Resource Advisor** expects a heartbeat signal from the dynamic agent. The default value is **200** seconds. After the maximum number of retries (specified in the **MissedHeartBeatCount** parameter) is exceeded, the **Resource Advisor** reports the related computer as unavailable. In a slow network, you might want to set this parameter to a higher value. However, defining a higher value might delay the updates on the availability status of computer systems. If, instead, you decrease this value together with the value defined for the **NotifyToResourceAdvisorPeriodSeconds** parameter, this might generate network traffic and increase CPU usage when updating cached data. The value defined in this parameter must be consistent with the **NotifyToResourceAdvisorPeriodSeconds** parameter defined in the `JobManager.ini` file, which defines the time interval for each dynamic agent to send the heartbeat signal to the **Resource Advisor**.

MissedHeartBeatCount

Specifies the number of missed heartbeat signals after which the computer is listed as not available. The default value is 2. In a slow network, you might want to set this parameter to a higher value.

MaxWaitingTime

Specifies the maximum time interval that a job must wait for a resource to become available. If the interval expires before a resource becomes available, the job status changes to Resource Allocation Failure. The default value is 600 seconds. You can override this value for each specific job by using the **Maximum Resource Waiting Time** parameter defined in the Job Brokering Definition Console. For more information about the **Maximum Resource Waiting Time** parameter, see the Job Brokering Definition Console online help. If you set this parameter to -1, no waiting interval is applied for the jobs. If you set this parameter to 0, the **Resource Advisor** tries once to find the matching resources and, if it does not find any resource, the job changes to the ALLOCATION FAILED status. If you increase this value, all submitted jobs remain in WAITING status for a longer time and the **Resource Advisor** tries to find matching resources according to the value defined for the **CheckInterval** parameter.

CheckInterval

Specifies how long the **Resource Advisor** waits before retrying to find matching resources for a job that did not find any resource in the previous time slot. The default value is 60 seconds.

TimeSlotLength

Specifies the time slot interval during which the **Resource Advisor** allocates resources to each job. Jobs submitted after this interval has expired are considered in a new time slot. The default value is 15 seconds. The default value is adequate for most environments and should not be modified. Setting this parameter to a higher value, causes the **Resource Advisor** to assign resources to higher priority jobs rather than to lower priority jobs when all jobs are trying to obtain the same resource. It might also, however, cause the job resource matching processing to take longer and the resource state updates from agents to be slowed down. Setting this parameter to a lower value, causes the **Resource Advisor** to process the resource matching faster and, if you have a high number of agents with frequent updates, to update the resource repository immediately. If job requirements match many resources, lower values ensure a better load balancing. If most jobs use resource allocation, do not lower this value because the allocation evaluation requires many processing resources.

NotifyTimeInterval

Specifies the interval within which the **Resource Advisor** retries to send notifications on the job status to the **Job Dispatcher** after a notification failed. The default value is 15 seconds. The default value is adequate for most environments and should not be modified.

MaxNotificationCount

Specifies the maximum number of attempts for the **Resource Advisor** to send notifications to the **Job Dispatcher**. The default value is 100. The default value is adequate for most environments and should not be modified.

ServersCacheRefreshInterval

Specifies with what frequency (in seconds) the Resource Advisor checks the list of active and backup dynamic workload broker servers for updates. This list is initially created when the master domain manager is installed, and after that it is updated every time a new backup master is installed and connected to the master domain manager database (the master domain manager and every backup master include also a dynamic workload broker server). When the Resource Advisor agents send their data about the resources discovered in each computer, they are able to automatically switch between the servers of this list, so that the dynamic workload broker server that is currently active can store this data in its Resource Repository. For this reason, the Resource Advisor agents must know at all times the list of all dynamic workload broker servers. The possible values range between 300 (5 minutes) and 43200 (12 hours). The default value is 600 seconds.

StatusCheckInterval

Specifies the time interval in seconds the Resource Advisor waits before polling for the status of a resource. For example this timeout applies when checking the status of a remote engine. The default value is 120 seconds.

After modifying the file, you must stop and restart WebSphere Application Server Liberty Base.

JobDispatcherConfig.properties file

The parameters in this file affect the following settings for the dynamic workload broker server installed on a master domain manager or dynamic domain manager:

- Encryption of any passwords sent in the JSDL definitions
- Time interval for retrying the operation after a **Job Dispatcher** failure
- Time interval for retrying the operation after a client notification failure
- Archive settings for job data
- Job history settings
- Gateways and dynamic workload broker server connection settings.

After modifying the file, you must stop and restart the IBM® WebSphere® server.

During the upgrade from version 8.5.1 the values you set for the properties for version 8.5.1 are preserved. The default values for the properties for version 8.6 are different from those in version 8.5.1. If you want to use the version 8.6 defaults, change them manually.

In the `JobDispatcherConfig.properties` file, the following parameters are available:

DatabaseCheckInterval

Specifies the time interval within which the dynamic workload broker server checks the availability of the database. The default value is **180** seconds.

EnablePasswordEncryption

Specifies that any user passwords contained in the JSDL definitions are to be encrypted when the definitions are sent to the agents. The default is `true`. Setting this property to `false` forces the dynamic workload broker server to send the passwords in plain text. This applies to any password field.

RAEndpointAddress

Specifies the URL of the **Resource Advisor**.

JDURL

Specifies the URL of the **Job Dispatcher**.

FailQInterval

Specifies the numbers of seconds for retrying the operation after the following failures:

- Client notification.
- Allocation, Reallocate, Cancel Allocation requests to **Resource Advisor**.
- Any database operation failed for connectivity reasons.

The default value is 30 seconds. Increasing this value improves recovery speed after a failure but can use many system resources if the recovery operation is complex. For example, if the workload broker workstation is processing a new Symphony file, this operation might require some time, so you should set this parameter to a high value. If you are not using workload broker workstation, this parameter can be set to a lower value.

MaxCancelJobAttemptsCount

The maximum number of times the Job Dispatcher attempts to cancel a shadow job or a job running on a dynamic agent when a request to kill the job is made and the kill request cannot be immediately processed. The default is 1440 attempts. The Job Dispatcher attempts to cancel the job every 30 seconds for a maximum number of times specified by this parameter.

MaxNotificationCount

Specifies the maximum number of retries after a client notification failure. The default value is 1440. For example, if the workload broker workstation is processing a new Symphony file, this operation might require some time, so you should set this parameter to a high value. If you are not using the workload broker workstation, this parameter can be set to a lower value.

MoveHistoryDataFrequencyInMins

Specifies how often job data must be deleted. The unit of measurement is minutes. The default value is 60 minutes. Increasing this value causes the **Job Dispatcher** to check less frequently for jobs to be deleted. Therefore, the volume of jobs in the **Job Repository** might increase and all queries might take longer to complete. Dynamic workload broker servers with high job throughput might require lower values, while low job throughputs might require higher values.

SuccessfulJobsMaxAge

Specifies how long successfully completed or canceled jobs must be kept in the **Job Repository** database before being archived. The unit of measurement is hours. The default value is 240 hours, that is ten days.

UnsuccessfulJobsMaxAge

Specifies how long unsuccessfully completed jobs or jobs in unknown status must be kept in the **Job Repository** database before being archived. The unit of measurement is hours. The default value is 720 hours, that is 30 days.

AgentConnectTimeout

Specifies the number of minutes that the dynamic workload broker server waits for a scheduling agent response after it first attempts to establish a connection with that agent. If the agent does not reply within this time, the server does not open the connection. Values range from 0 to 60 (use 0 to wait indefinitely). The default is 3.

AgentReadTimeout

Specifies the number of minutes that the dynamic workload broker server waits to receive data from established connections with a scheduling agent or a gateway. If no data arrives within the specified time, the server closes the connection with the agent. Values range from 0 to 60 (use 0 to wait indefinitely). The default is 3.

GatewayPollingTimeout

Add this parameter to specify the number of minutes that the gateway waits to receive data from established connections with a dynamic workload broker. If no data arrives within the specified time, the gateway closes the connection with the dynamic workload broker. Values range from 1 to 60. The default is 1 minute.

GatewayConnectionTimeout

Add this parameter to specify the number of seconds that the dynamic workload broker server waits for a gateway receiving data after the dynamic workload broker first attempts to send data to the gateway. If the gateway does not reply within this time, the dynamic workload broker does not open the connection. Values range from 1 to 60. The default is 10 seconds.

MaxNumberOfParallelGateways

Add this parameter to specify the number of gateways that dynamic workload broker server can manage without lack of performances. Values range from 3 to 100. The default is 3.

**Note:**



If an unexpected job workload peak occurs and a cleanup of the database is required earlier than the value you specified in the `MoveHistoryDataFrequencyInMins` parameter, you can use the `movehistorydata` command to perform a cleanup before the scheduled cleanup is performed.

BrokerWorkstation.properties file

If you need to make configuration changes to the broker server after the installation has completed, you can edit the `BrokerWorkstation.properties` file. The `BrokerWorkstation.properties` file contains the following configuration properties:

DomainManager.Workstation.Name

The name of the domain manager workstation.

DomainManager.Workstation.Port

The port of the domain manager workstation.

MasterDomainManager.Name

The name of the master domain manager workstation.

Broker.Workstation.Name

The name of the broker server in the IBM Workload Scheduler production plan. This name is first assigned at installation time.

MasterDomainManager.HostName

The host name of the master domain manager workstation.

MasterDomainManager.HttpsPort

The HTTPS port of the master domain manager workstation.

Broker.Workstation.Port

The port used by the broker server to listen to the incoming traffic (equivalent to the Netman port). It is first assigned at installation time. This port number must always be the same for all the broker servers that you define in your IBM Workload Scheduler network (one with the master domain manager and one with every backup master you install) to ensure consistency when you switch masters.

DomainManager.Workstation.Domain

The name of the domain where the broker server is registered.

Broker.AuthorizedCNs

The list of prefixes of common names authorized to communicate with the broker server.

Broker.CertificateExpirationInterval

The number of days before the certificate on the agent expires. During this interval, the certificate is set in expiring status and the agent tries to download a new version of the certificate from the master domain manager, if available. Supported values are any integer greater than zero. The default value is 15 days.

Broker.Workstation.Enable

A switch that enables or disables the broker server. The value can be `true` or `false`. The default value is `true`.

Set this value to `false` if you decide not to use a broker server. Not using the broker server means that you can submit jobs dynamically on the dynamic workload broker directly (using either the Dynamic Workload Console or the dynamic workload broker command line) without using the scheduling facilities of IBM Workload Scheduler.

Broker.Workstation.CpuType

The workstation type assigned to the broker server. Supported values are:

- master domain manager (master)
- backup master domain manager (fta)
- dynamic domain manager (fta, broker, agent)
- backup dynamic domain manager (fta, broker, agent)

Broker.Workstation.RetryLink

The number of seconds between consecutive attempts to link to the broker server. The default is 600.

Note that no SSL security is available for the connection between the master domain manager and the broker server. All the data between the two workstations is sent unencrypted. If this might cause a security risk in your environment, you can choose not to use the broker server functions, by setting `Broker.Workstation.Enable` to `false`.

If you need to modify the event processor server, for example to use a load balancer, add the following two keywords in the file:

Broker.Workstation.evtproc.previous_hostname=new_hostname

Specify the previous hostname and the new hostname of the event processor.

Broker.Workstation.evtproc.previous_port=new_port

Specify the previous and new port of the event processor.

After stopping and restarting WebSphere Application Server Liberty Base, the dynamic domain manager sends the updated information to the dynamic agents.

Archiving job data

Job definitions created using the Dynamic Workload Console are stored in the **Job Repository** database. The **Job Repository** database stores also the jobs created when the job definitions are submitted to the dynamic workload broker.

Job information is archived on a regular basis. By default, successful jobs are archived every 24 hours. Jobs in failed or unknown status are archived by default every 72 hours.

You can configure the time interval after which job data is archived using the following parameters:

- **MoveHistoryDataFrequencyInMins**
- **SuccessfulJobsMaxAge**
- **UnsuccessfulJobsMaxAge**

These parameters are available in the `JobDispatcherConfig.properties` file, as described in [JobDispatcherConfig.properties file on page 101](#). You can also use the `movehistorydata` command to perform a cleanup before the scheduled cleanup is performed.

Configuring to schedule J2EE jobs

About this task

Using the dynamic workload broker component you can schedule J2EE jobs. To do this you must complete the following configuration tasks:

- [Configure the J2EE executor on page 106](#) on every agent on which you submit J2EE jobs.
- [Configure the J2EE Job Executor Agent on page 111](#) on an external WebSphere® Application Server

Configuring the J2EE executor

About this task

To dynamically schedule J2EE jobs, you must configure the following property files on every agent on which you submit J2EE jobs:

- `J2EEJobExecutorConfig.properties`
- `logging.properties`
- `soap.client.props`

These files are configured with default values at installation time. The values that you can customize are indicated within the description of each file.

J2EEJobExecutorConfig.properties file

Use the `J2EEJobExecutorConfig.properties` file to configure the J2EE executor

The file is located in:

On Windows operating systems

```
TWA_home>\JavaExt\version_number>\cfg
```

On UNIX operating systems

```
TWA_DATA_DIR>/JavaExt/cfg
```

The keywords of this file are described in the following table:

Table 22. J2EEJobExecutorConfig.properties file keywords

Keyword	Specifies...	Default value	Must be customized
wasjaas.default	The path to the WebSphere Application Server Liberty Base configuration file (<code>wsjaas_client.conf</code>) used to authenticate on the external WebSphere® Application Server using JAAS security.	<code>TWA_home/TWS/JavaExt/cfg/wsjaas_client.conf</code> or <code>TWA_home\TWS\JavaExt\cfg\wsjaas_client.conf</code>	Optionally yes, if you move the file to the path you specify.
credentials.mycred	The credentials (ID and password) used to establish the SOAP connection to the external WebSphere® Application Server when using indirect scheduling (the password must be {xor} encrypted)	<code>wasadmin,{xor}KD4sPjsyNjE\=</code> (ID= <code>wasadmin</code> and password= <code>wasadmin</code> in {xor} encrypted format)	Yes, see WebSphere Application Server Liberty Base documentation, for example securityUtility command to learn how to encrypt your password.
connector.indirect	The name of the communication channel with WebSphere® Application Server. Selecting an indirect invoker means that dynamic workload broker uses an existing WebSphere® Application Server scheduling infrastructure that is already configured on a target external WebSphere® Application Server.	A single line with the following values separated by commas: <ul style="list-style-type: none"> • <code>indirect</code> keyword • Name of the scheduler: <code>sch/MyScheduler</code> • <code>soap</code> keyword • Host name of the external WebSphere® Application Server instance: <code>washost.mydomain.com</code> • SOAP port of the WebSphere® Application Server instance: <code>8880</code> • Path to the <code>soap.client.props</code> file: <code>TWA_home/TWS/JavaExt/cfg/soap.client.props</code> • Credentials keyword: <code>mycred</code> 	You must customize the following: <ul style="list-style-type: none"> • The scheduler name. Replace the <code>sch/MyScheduler</code> string with the JNDI name of the IBM® WebSphere® scheduler that you plan to use. • The host name of the external WebSphere® Application Server instance. • The SOAP port of the external WebSphere® Application Server instance.

Table 22. J2EEJobExecutorConfig.properties file keywords (continued)

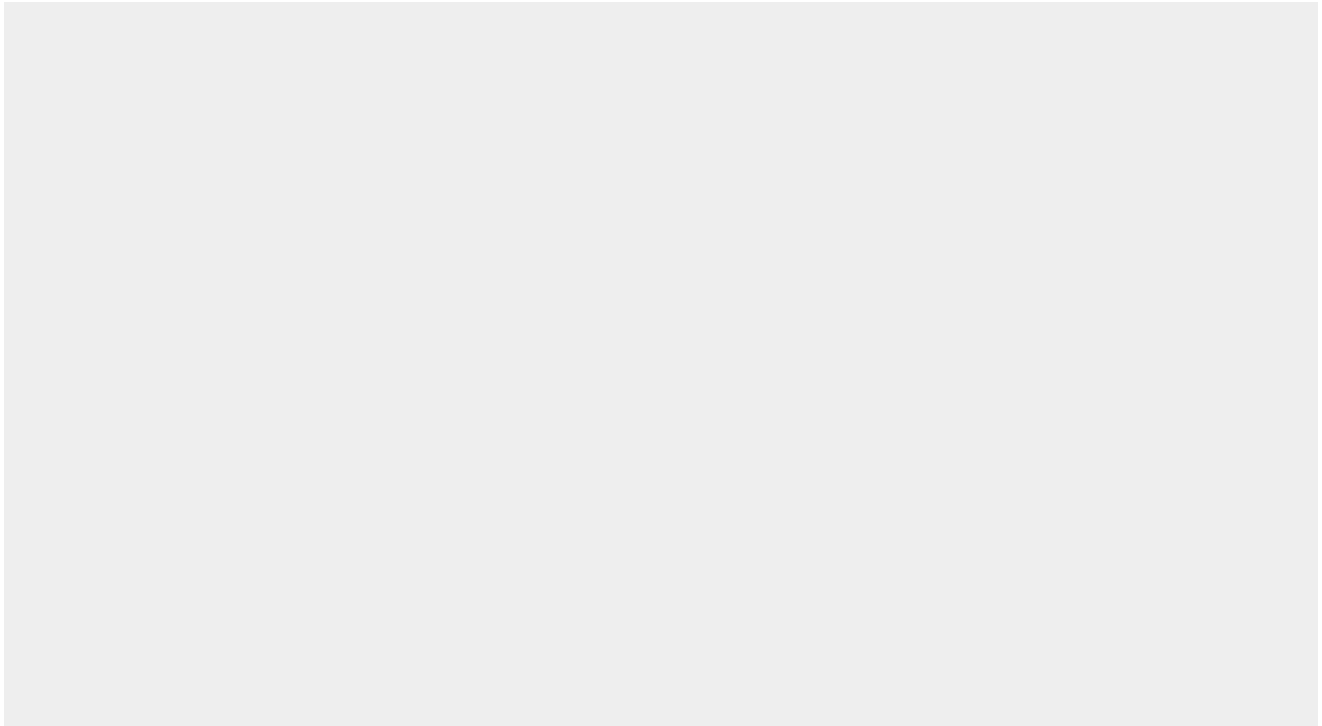
Keyword	Specifies...	Default value	Must be customized
connector.direct	The name of the direct communication channel without using the WebSphere® Application Server scheduler. Select a direct invoker to have dynamic workload brokerimmediately forward the job to the external WebSphere® Application Server instance components (EJB or JMS). When creating the job definition, you can specify if you want to use a direct or indirect connector in the J2EE Application pane in the Application page in the Job Brokering Definition Console, or in the invoker element in the JSDL file. For more information about the Job Brokering Definition Console, see the online help.	A single line with the following values separated by commas: <ul style="list-style-type: none"> • <code>direct</code> keyword • The following string: <pre>com.ibm.websphere.naming.WsnInitialContextFactory</pre> • The following string: <pre>corbaloc:iiop:washost.mydomain.com:2809</pre> 	You must customize the following: <ul style="list-style-type: none"> • The host name of the external WebSphere® Application Server instance: <code>washost.mydomain.com</code> • The RMI port of the external WebSphere® Application Server instance: <code>2809</code>
trustStore.path	The path to the WebSphere® Application Server trustStore file (this file must be copied to this local path from the WebSphere® Application Server instance).	<code>TWA_home/TWS/JavaExt/cfg/DummyClientTrustFile.jks</code>	You can change the path (<code>TWA_home/TWS/JavaExt/cfg</code>), if you copy the trustStore path from the external WebSphere® Application Server to this path.
trustStore.password	The password for the WebSphere® Application Server trustStore file.	WebAs	Yes

The logging.properties file

About this task

The path to this file is `TWA_home/TWS/JavaExt/cfg/logging.properties` (`TWA_home\TWS\JavaExt\cfg\logging.properties`) on the agent.

After installation, this file is as follows:



You can customize:

- The logging level (from INFO to WARNING, ERROR, or ALL) in the following keywords:

`.level`

Defines the logging level for the internal logger.

`com.ibm.scheduling`

Defines the logging level for the job types with advanced options. To log information about job types with advanced options, set this keyword to ALL.

- The path where the logs are written, specified by the following keyword:

`java.util.logging.FileHandler.pattern`

The soap.client.props file

About this task

The path to this file is as follows:

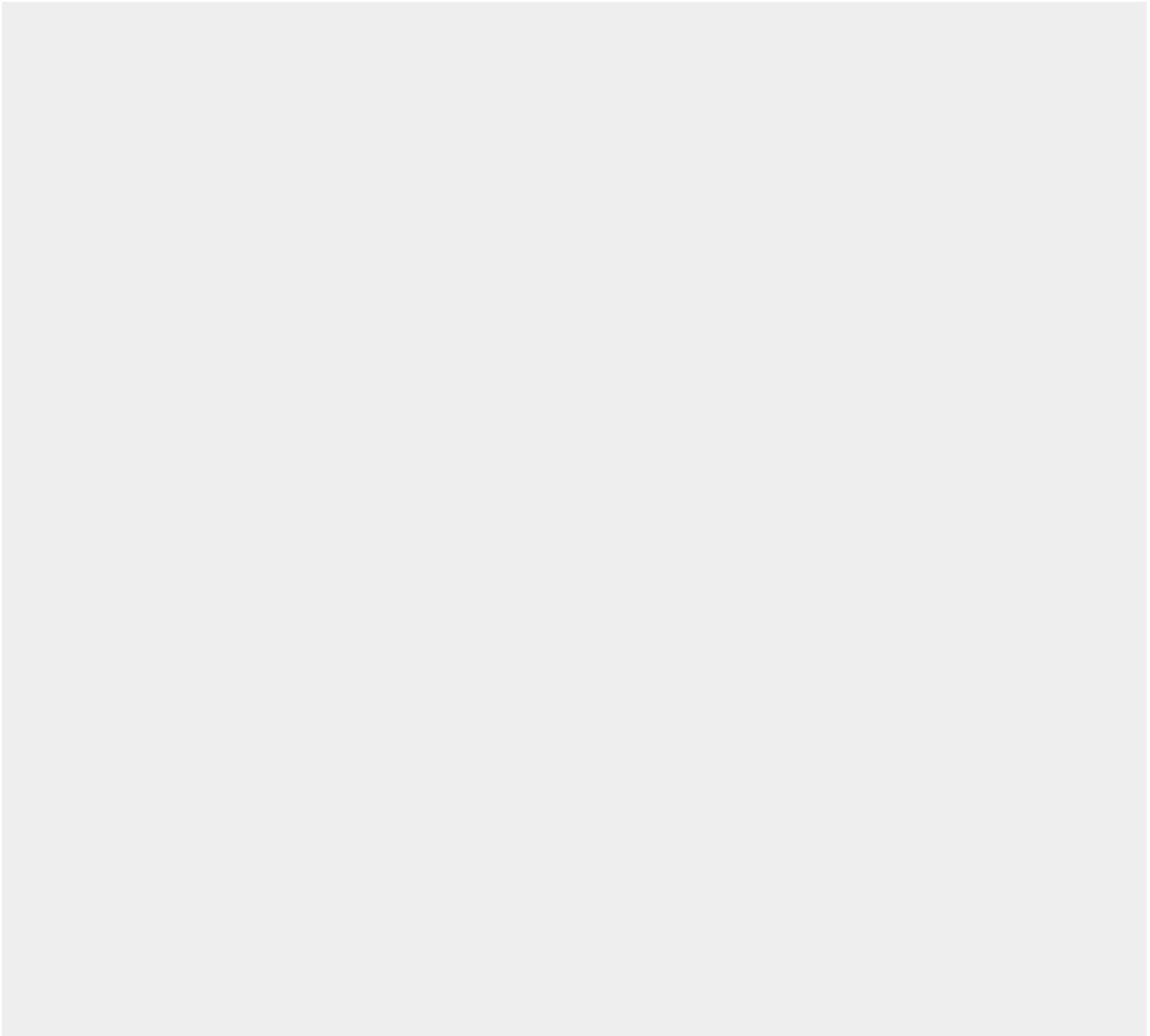
On Windows operating systems

`<TWA_home>\TWS\JavaExt\cfg\soap.client.props`

On UNIX operating systems

`<TWA_DATA_DIR>/JavaExt/cfg/soap.client.props`

After installation, this file is as follows:



If you want to enable SOAP client security, you must:

1. Change `com.ibm.SOAP.securityEnabled` to `true`
2. Customize:
 - `com.ibm.SOAP.loginUserId` with the true WebSphere Application Server Liberty Base administrator user ID.
 - `com.ibm.SOAP.loginPassword` with the true WebSphere Application Server Liberty Base administrator password in {xor} encrypted format. For further details about encrypting and decrypting passwords in xor format, see the documentation, for example [WebSphere {xor} password decoder and encoder](#).

Configuring the J2EE Job Executor Agent

About this task

To set up the environment on the external WebSphere® Application Server, Version 7.0 for the J2EE Job Executor Agent, do the following:

Create a Service Integration Bus

1. Open the WebSphere® Administrative Console (for example, `http://localhost:9060/admin`, depending on the admin port you configured).
2. Expand **Service Integration** and select **Buses**. The Buses window is displayed.
3. Click **New** to display the Buses configuration window.
4. Type a name for the new bus, for example **MyBus** and click **Next** and then **Finish** to confirm.
5. Click the MyBus name and the MyBus properties are displayed.
6. Under Topology, click **Bus Members**. The Buses→MyBus→Bus members window is displayed.
7. Click **Add**, select the **Server** radio button, choose **your_application_server_name**, click **Next**, and then click **Finish**.
8. When the *Confirm the addition of a new bus member* panel is displayed, click **Finish**.
9. Select **Service Integration → Buses → MyBus → Destinations → New**.
10. Select **Queue** as the type and click **Next**
11. Type **BusQueue** as the identifier and assign the queue to a bus member. Click **Next**. In the confirmation panel click **Finish**.

Configure the Default Messaging Service

1. From the left panel of the WebSphere® Administrative Console, expand **Resources**→**JMS**→**JMS Providers**, then click **Default messaging** at the server level as scope.
2. In the **Connection Factories** section, click **New**.
3. On the New JMS connection factory window, type in the following fields:

Name

MyCF

JNDI name

jms/MyCF

Bus name

MyBus

Provider endpoints

<hostname>:<Basic SIB port number>:BootstrapBasicMessaging;<hostname>:<Secure SIB port number>:BootstrapSecureMessaging

4. Select again **Resources** → **JMS** → **JMS Providers** → **Default Messaging** at the server level as scope, locate the section **Destinations**, and click **Queues**. Click **New** and type in the following fields as shown:

Name=MyQueue
JNDI name=jms/MyQueue
Bus name=MyBus
Queue name=BusQueue

Click **Ok**.

5. Select again **Resources** → **JMS** → **JMS Providers** → **Default Messaging** at the server level as scope, and locate the section **Activation Specifications**.

6. Click **JMS activation specification**. Click **New** and type in the following fields as shown:

Name=MyActSpec
JNDI name=eis/MyActSpec
Bus name=MyBus
Destination type=Queue
Destination JNDI name=jms/MyQueue

Click **Ok**.

Configure the Java security

1. Select **Security** → **Secure Administration, applications and infrastructure**.
2. Locate the **Authentication** section, expand the **Java Authentication and Authorization Service**, and click **J2C authentication data**.

3. Click **New** and type in the following fields as shown:

Alias=*usr*
User ID=*usr*
Password=*pwd*

where *usr* is the user ID authenticated when using connector security and *pwd* is the related password.

4. Click **Ok**.

Create an XA DataSource

1. In the left pane, go to **Resources** → **JDBC** . . → **JDBCProviders**. In the resulting right pane, check that the scope is pointing to **your_application_server_name**.
2. Locate the **DERBY JDBC Provider (XA)** entry and click it.
3. Locate the **Additional Properties** section and click **Data Sources**.

4. Click **New** and type in the following fields as shown:

Name = MyScheduler XA DataSource
JNDI name = jdbc/SchedulerXADS
Database name = \${USER_INSTALL_ROOT}/databases/Schedulers/\${SERVER}/SchedulerDB;create=true

5. At the top of the page, click **Test connection button**.

6. Even if you get a negative result, modify the **Database name** field, deleting the part `;create=true`. Click **Ok**.

Create a WorkManager

1. In the left pane, go to **Resources** → **Asynchronous beans** → **Work managers** and click **New**.
2. type in the following fields as shown:
 - Name=SchedulerWM
 - JNDI name=wm/SchedulerWM
3. Click **Ok**.

Create and configure a scheduler

1. In the left pane, go to **Resources** → **Schedulers** and click **New**.
2. type in the following fields as shown:
 - Name=MyScheduler
 - JNDI name=sch/MyScheduler
 - Data source JNDI name=jdbc/SchedulerXADS
 - Table prefix=MYSCHED
 - Work managers JNDI name=wm/SchedulerWM
3. Click **Ok**.
4. Select **MyScheduler** and click **Create tables**.
5. Deploy the test application.

Security order of precedence used for running J2EE tasks

There are three ways of verifying that a task runs with the correct user credentials. Tasks run with specified security credentials using the following methods:

1. Java™ Authentication and Authorization Service (JAAS) security context on the thread when the task was created.
2. `setAuthenticationAlias` method on the `TaskInfo` object.
3. A specified security identity on a `BeanTaskInfo` task `TaskHandler` EJB method.

The authentication methods are performed in the order listed above, so that if an authentication method succeeds, the following checks are ignored. This means that the `usr` and `pwd` credentials defined in **Configure the Java™ security** take precedence over any credentials specified in the tasks themselves.


Configuring to schedule job types with advanced options

About this task

You can define job types with advanced options by using the related configuration files. The options you define in the configuration files apply to all job types with advanced options of the same type. You can override these options when defining the job by using the Dynamic Workload Console or, if you are in a distributed environment, the **composer** command.

Configuration files are available on each dynamic agent in `TWA_home/TWS/JavaExt/cfg` for the following job types with advanced options:

Table 23. Configuration files for job types with advanced options

Job type	File name	Keyword
<ul style="list-style-type: none"> • Database job type • MSSQL Job 	DatabaseJobExecutor.properties	<p>Use the <code>jdbcDriversPath</code> keyword to specify the path to the JDBC drivers. Define the keyword so that it points to the JDBC jar files directory, for example:</p> <pre>jdbcDriversPath=c:\mydir\jars\jdbc</pre> <p>The JDBC jar files must be located in the specified directory or its subdirectories. Ensure you have list permissions on the directory and its sub subdirectories.</p> <p> Note: For the MSSQL database, use version 4 of the JDBC drivers.</p>
Java™ job type	JavaJobExecutor.properties	<p>Use the <code>jarPath</code> keyword to specify the path to the directory where the jar files are stored. This includes all jar files stored in the specified directory and all sub directories.</p>
J2EE job type	J2EEJobExecutorConfig.properties	<p>For more information about the J2EE job type, see the topic about configuring to schedule J2EE jobs in the <i>IBM Workload Scheduler: Administration Guide</i>.</p>

Configuring security roles for users and groups

The dynamic workload broker provides two commands for managing resources and job definitions:

resource

to create, modify, associate, query, or set resources online or offline. For more information, see the section about the resource command in *User's Guide and Reference*.

jobstore

to manage job definitions. For more information, see the section about the jobstore command in *Scheduling Workload Dynamically*.

At master domain manager installation time, the `broker_role_mapping.xml` template is created to configure in WebSphere Application Server Liberty Base the users and groups authorized to use the dynamic workload broker commands. For the configuration procedure, see [Mapping security roles to users and groups in WebSphere Application Server Liberty Base on page 115](#).

Mapping security roles to users and groups in WebSphere Application Server Liberty Base

About this task

When the dynamic workload broker instance is installed on your master domain manager, corresponding roles are set up in WebSphere Application Server Liberty Base. By default, these roles are not used. However, the authorization required to perform any tasks is always validated by WebSphere Application Server Liberty Base. Users are required to provide credentials for managing resources and job definitions using the resource and jobstore commands. These credentials correspond to existing users defined in the domain user registry or the LDAP server.

To allow users and groups to access the dynamic workload broker functions, they must be mapped to the security roles in WebSphere Application Server Liberty Base. This mapping allows those users and groups to access applications defined by the role. At installation time, the IBM® Workload Scheduler administrative user (**wauser**) is assigned the **Administrator** role in WebSphere Application Server Liberty Base. The following roles are also created but they are not assigned to any users nor groups:

Operator

Monitors and controls the jobs submitted.

Administrator

Manages the scheduling infrastructure.

Submitter

Manages the submission of their own jobs and monitors and controls the job lifecycle. This is the typical role for an IBM Workload Scheduler user.

IBM Workload Scheduler acts as submitter of jobs to the IBM Workload Scheduler dynamic agent.

Configurator

Is the entity responsible for running the jobs on a local environment.

To map security roles to users and groups on the WebSphere Application Server Liberty Base, edit the `broker_role_mapping.xml` file located in `<Liberty_installation_directory>/usr/servers/engineServer/configDropins`.

You can edit the file to associate users and groups to the **Operator**, **Administrator**, **Developer**, or **Submitter** roles, as follows:

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.

To enable all users to use the dynamic workload broker commands, uncomment the `special-subject` string, otherwise, specify the list of users or groups for each role.

Examples

In the following example, the **Operator** role is associated to user **user1**, the **Submitter** role is associated to all users belonging to **group1**, and the **Configurator** role is associated to all users authenticated by the server:

```
<server>
  <enterpriseApplication id="SchedulerEAR">
    <application-bnd>
      <security-role id="adminRole" name="Administrator">
        <user access-id="${user.twsuser.id}" name="${user.twsuser.id}" />
        <run-as userid="${user.twsuser.id}" password="${user.twsuser.password}"/>

      </security-role>
      <security-role id="operatorRole" name="Operator">
        <user name=?user1?/>

      </security-role>
      <security-role id="submitterRole" name="Submitter">
        <group name=?group1?/>

      </security-role>
      <security-role id="configuratorRole" name="Configurator">
        <special-subject type="ALL_AUTHENTICATED_USERS"/>
      </security-role>
    </application-bnd>
  </enterpriseApplication>
```

broker_role_mapping.xml file

Example

```
<server>
  <enterpriseApplication id="SchedulerEAR">
    <application-bnd>
      <security-role id="adminRole" name="Administrator">
        <user access-id="${user.twsuser.id}" name="${user.twsuser.id}" />
        <run-as userid="${user.twsuser.id}" password="${user.twsuser.password}"/>

      </security-role>
      <security-role id="operatorRole" name="Operator">

      </security-role>
      <security-role id="submitterRole" name="Submitter">

      </security-role>
      <security-role id="configuratorRole" name="Configurator">

      </security-role>
    </application-bnd>
  </enterpriseApplication>
```

Configuring command-line client access authentication

This section describes how to reconfigure the connection used by the command line client.

The command line client is installed automatically on the master domain manager. On the master domain manager you use it to run all of the commands and utilities.

On any other workstation you use it to run one of the following commands:

- **evtdef**
- **composer**
- **optman**
- **planman**
- **sendevent**

It is configured automatically by the installation process, but if you need to change the credentials that give access to the server on the master domain manager, or you want to use it to access a different master domain manager, modify the **connection parameters** as described in [Connection parameters on page 117](#).



Note:

1. The **connection parameters** are not required to use the local **conman** program on a fault-tolerant agent.
2. The command-line client on the master domain manager uses exactly the same mechanism to communicate with the server as it does when it is installed remotely.

Connection parameters

About this task

The connection parameters can be provided in one of three ways:

Define them in **localopts**

All fields except *username* and *password*, can be defined by editing the `TWA_home/TWS/localopts` properties file on the computer from which the access is required. See [Setting local options on page 51](#) for a full description of the file and the properties.

In **localopts** there is a section for the general connection properties, which contains the following:

```

host = host_name
protocol = protocol
port = port number
proxy = proxy server
proxyport = proxy server port number
timeout = seconds
defaultws = master_workstation
useropts = useropts_file

```

In addition, there are separate groups of SSL parameters which differ depending on whether your network is FIPS-compliant, and thus uses GSKit for SSL, or is not, and uses OpenSSL (see [FIPS compliance on page 330](#) for more details):

FIPS-compliant (GSKit)

```
CLI SSL keystore file = keystore_file_name
CLI SSL certificate keystore label = label
CLI SSL keystore pwd = password_file_name
```

Not FIPS-compliant (OpenSSL)

```
CLI SSL server auth = yes|no
CLI SSL cipher = cipher_class
CLI SSL server certificate =certificate_file_name
CLI SSL trusted dir =trusted_directory
```

Store some or all of them in useropts

As a minimum, the **username** and **password** parameters can be defined in the `user_home/.TWS/useropts` file for the user who needs to make the connection. Also, if you need to personalize for a user any of the properties normally found in the `localopts` file, add the properties to the `useropts` file. The values in the `useropts` file always take precedence over those in the `localopts` file. See [Setting user options on page 76](#) for a full description of the file and the properties.

The minimum set of properties you would find in **useropts** is as follows:

```
username=user_ID
password=password
```

Supply them when you use the command

When you use any of the commands you can add one or more of the connection parameters to the command string. These parameters take precedence over the parameters in **localopts** and **useropts**. This allows you, for example, to keep the parameters in the **localopts** file and just get users to supply the **username** and **password** parameters when they use one of the commands, avoiding the necessity to store this data in the **useropts** file for each user..

The parameters can either be supplied fully or partially in a file, to which you refer in the command string, or typed directly as part of the command string. The full syntax is as follows:

```
[-file <parameter_file>
|
-host <host_name>]
[-password <user_password>]
[-port <port_number>]
[-protocol {http|https}]
[-proxy <proxy_name>]
[-proxyport <proxy_port_number>]
[-timeout <timeout>]
[-username <username>]
```

-file <parameter_file>

A file containing one or more of the connection parameters. Parameters in the file are superseded if the corresponding parameter is explicitly typed in the command.

-host <host_name>

The host name or IP address of the master domain manager to which you want to connect.

-password <user_password>

The password of the user supplied in the `-username` parameter.

-port <port_number>

The listening port of the master domain manager to which you want to connect.

-protocol {http|https}

Enter either http or https, depending on whether you want to make a secure connection.

-proxy <proxy_name>

The host name or IP address of the proxy server involved in the connection (if any).

-proxyport <proxy_port_number>

The listening port of the proxy server involved in the connection (if any).

-timeout <timeout>

The number of seconds the command line client is to wait to make the connection before giving a timeout error.

-username <username>

The user ID of the user making the connection.

**Note:**

From the command line, neither the default workstation, nor the command line client SSL parameters can be supplied. These must always be supplied in either the `localopts` (see [Setting local options on page 51](#)) or the `useropts` file for the user (see [Setting user options on page 76](#)).

For monitoring commands, such as **conman showjobs**, **conman showresources**, and so on, IBM® Workload Scheduler uses the connection parameters of the user logged on to the computer

The command line client needs to assemble a full set of parameters, and it does so as follows:

1. First it looks for values supplied as parameters to the command
2. Then, for any parameters it still requires, it looks for parameters supplied in the file identified by the `-file` parameter
3. Then, for any parameters it still requires, it looks in the `useropts` file for the user
4. Finally, for any parameters it still requires, it looks in the `localopts` file

If a setting for a parameter is not specified in any of these places an error is displayed.

Entering passwords

About this task

Password security is handled as follows:

Password entered in `useropts` file

You type the connection password into the `useropts` file in unencrypted form. When you access the interface for the first time it is encrypted. This is the preferred method.

Password entered in the parameter file used by the command

You type the connection password into the parameter file in unencrypted form. It is not encrypted by using the command. Delete the file after use to ensure password security.

Password entered using the `-password` parameter in the command

You type the password in the command string in unencrypted form. It remains visible in the command window until you clear the command window contents.



Note: On Windows™ workstations, when you specify a password that contains double quotation marks (") or other special characters, make sure that the character is escaped. For example, if your password is `tws11"tws`, write it as `"tws11\"tws"` in `useropts`.

An active-active high availability scenario

Implement active-active high availability between the Dynamic Workload Console and the master domain manager so that a switch to a backup is transparent to Dynamic Workload Console users.

Use a load balancer between the Dynamic Workload Console servers and the master domain manager so that in the event the master needs to switch to a backup, the switch is transparent to console users.

Configure the master domain manager and backup master domain managers behind a second load balancer so that the workload is balanced across all backup master domain managers and the master domain manager. Load balancing distributes workload requests across all configured nodes to avoid any single node from being overloaded and avoids a single point of failure.

You might already have installed and configured a number of backup master domain managers, in addition to your master domain manager, that you use for dedicated operations and to alleviate the load on the master domain manager. For example, you might have one dedicated to monitoring activities, another for event management, and perhaps your scheduling activities are run on the master domain manager. Administrators must create engine connections for each of these and users have to switch between engines to run dedicated operations. Should one of them go down, users need to be notified about which engine to use as a replacement and switch to the replacement engine.

To simplify this, configure a load balancer in front of the master domain manager and backup master domain managers so that users are unaware of when a switch occurs and administrators configure a single engine connection in single-sign on that points to the name or IP address and port number of the load balancer and not ever need to know the hostname of the

current active master. The load balancer monitors the engine nodes and takes over the task of balancing the workload and the switch to a backup master domain manager is completely transparent to console instance users. Any backup master domain manager can satisfy HTTP requests, even those that can be satisfied only by the active master, such as requests on the plan, because the requests are proxied to and from the active master.

To complete the picture of a full high availability IBM® Workload Scheduler environment, the RDBMS and the Dynamic Workload Console need to be configured in high availability. If your RDBMS includes a high availability disaster recovery (HADR) feature and it is enabled, you can configure the `datasource.xml` file on the WebSphere Application Server Liberty Base server of the master and backup components to add failover properties. The key-value pairs to set depend on your specific RDBMS. As an example, Db2®'s datasource can be configured with the following set of properties in the XML element named **properties.db2.jcc**:

```
<properties.db2.jcc
  databaseName="TWS"
  user="..."
  password="..."
  serverName="MyMaster"
  portNumber="50000"
  clientRerouteAlternateServerName="MyBackup"
  clientRerouteAlternatePortNumber="50000"
  retryIntervalForClientReroute="3000"
  maxRetriesForClientReroute="100"
/>
```

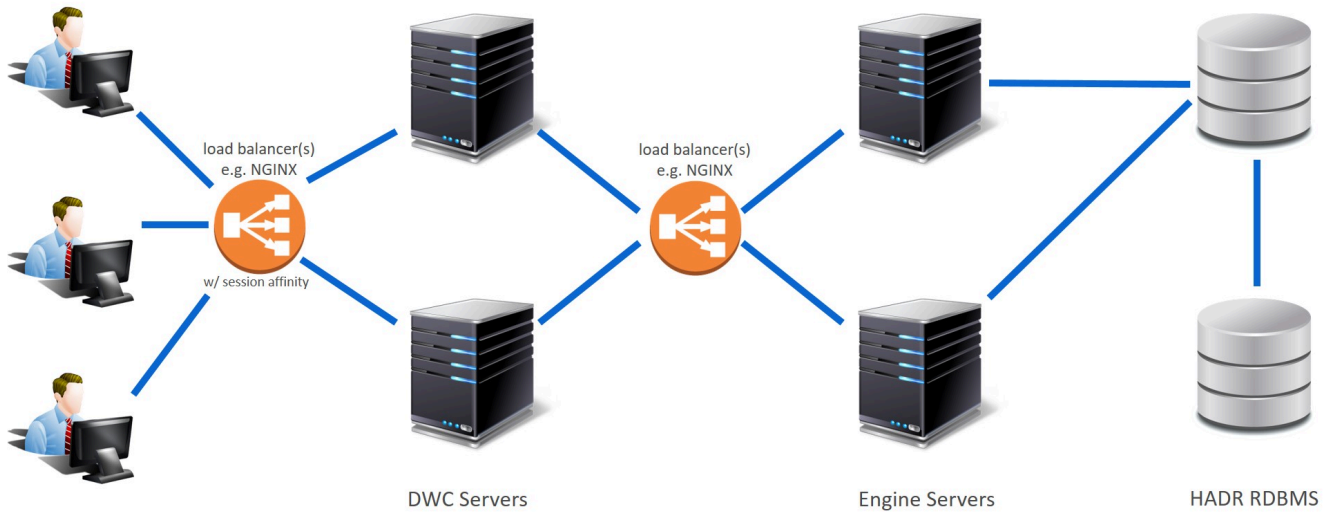
For the Dynamic Workload Console, ensure that it is connected to an external database (not the embedded Apache Derby database), replicate it, and link the consoles to a load balancer that supports session affinity so that requests related to the same user session are dispatched to the same console instance.



Note: If your environment includes the IBM Workload Scheduler Agent for z/OS to schedule jobs (JCL) on the JES2 subsystem of z/OS, ensure that the value for the **host.bootstrap.port.sec** parameter specified in the `ports_variables.xml` file is the same on every workstation hosting the Dynamic Workload Console component in your environment. For more information about the location of the `ports_variables.xml` file, see [Configuring IBM Workload Scheduler using templates on page 428](#).

The following is a sample of the high-level architecture of an active-active high availability system with two load balancers

Figure 1. An end-to-end environment configured for high availability



This environment configuration offers numerous benefits:

High availability

The load balancer monitors the nodes and takes care of balancing the workload across the nodes, eliminating the possibility of a node creating a single point of failure.

Scalability

As client requests increase, you can add additional backup master domain manager nodes to the configuration to support the increased workload.

User-friendly

Users are unaware of when a switch occurs to a different node. Administrators do not have to worry about creating new engine connections to run workloads on a different node.

Low overhead

This configuration does not require any manual intervention when nodes become unavailable. Additional flexibility is provided to console instance users who no longer have to run certain operations on dedicated nodes.

Optimization of hardware

Load balancing distributes session requests across multiple servers thereby utilizing hardware resources equally.



Note: When there are multiple Dynamic Workload Console servers connected to a load balancer and one of the servers becomes unavailable, load balancing takes place automatically, however, console users need to perform a page refresh and reopen any tabbed pages that were previously opened.

The following is intended to be a high-level view of the steps required to implement a scenario of this kind. It is an example and is not meant to be a verified procedure.

1. Install a load balancer on a workstation either within the IBM® Workload Scheduler environment or external to the environment. Ports must be open between the load balancer and the Dynamic Workload Console nodes and the engine nodes.
2. Configure multiple Dynamic Workload Console instances in a cluster where the consoles share the same repository settings and a load balancer takes care of dispatching and redirecting connections among the nodes in the cluster. The load balancer must support **session affinity**. See the topic about configuring high availability across multiple Dynamic Workload Console nodes.
3. Exchange certificates between the load balancer and the Dynamic Workload Console and between the second load balancer and the master domain manager and backup master domain manager nodes.
4. Configure the load balancer configuration file with details about the Dynamic Workload Console, master domain manager, and backup master domain managers. The configuration file indicates which nodes (Dynamic Workload Console, master domain manager, and backup master domain managers) are available and the routes to be used to dispatch client calls to the Dynamic Workload Console server nodes.
5. Configure an engine connection that points to the name or IP address of the load balancer, and specify the incoming port number to the load balancer that corresponds to the outgoing port number to the master (default port number 31116). The load balancer must point to the HTTPS of the Dynamic Workload Console and the HTTPS of the master domain manager.
6. Configure an RDBMS in high availability and enable the HADR feature.
7. To configure the Dynamic Workload Console nodes in a cluster behind the load balancer, modify the `ports_config.xml` file as follows:

```
<httpEndpoint host="{httpEndpoint.host}" httpPort="{host.http.port}" httpsPort="{host.https.port}"
  id="defaultHttpEndpoint">
  <httpOptions removeServerHeader="true"/>
  <remoteIp useRemoteIpInAccessLog="true"/>
</httpEndpoint>
```

This solution requires a load balancer that supports session affinity. Nginx is an example of a load balancer of this kind. The following is an abstract from a configuration file for an Nginx load balancer that demonstrates the configuration settings necessary to implement the high availability use case depicted in [Figure 1: An end-to-end environment configured for high availability on page 122](#).

```
user nginx;
worker_processes 10; ## Default: 1
worker_rlimit_nofile 8192;

error_log /var/log/nginx/error.log warn;
pid /var/run/nginx.pid;

events {
  worker_connections 4096; ## Default: 1024
}

http {
  include /etc/nginx/mime.types;
  default_type application/octet-stream;
```

```

log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                '$status $body_bytes_sent "$http_referer" '
                '"$http_user_agent" "$http_x_forwarded_for"';

access_log /var/log/nginx/access.log main;

sendfile      on;

keepalive_timeout 65;

upstream wa_console { ##DWC configuration
    ip_hash;
    server DWC1_HOSTNAME:DWC1_PORT max_fails=3 fail_timeout=300s;
    server DWC2_HOSTNAME:DWC2_PORT max_fails=3 fail_timeout=300s;
    keepalive 32;
}

upstream wa_server_backend_https {
    server MDM1_HOSTNAME:MDM1_PORT weight=1;
    server MDM2_HOSTNAME:MDM2_PORT weight=1;
}

server{
    listen      443 ssl;

    ssl_certificate /etc/nginx/certs/nginx.crt;
    ssl_certificate_key /etc/nginx/certs/nginxkey.key;
    ssl_trusted_certificate /etc/nginx/certs/ca-certs.crt;
    location /
    {
        proxy_pass https://wa_console;
        proxy_cache off;

        proxy_set_header Host $host;

        proxy_set_header Forwarded " $proxy_add_x_forwarded_for;proto=$scheme";
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header X-Forwarded-Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-Port 443;
    }
}

server{
    listen      9443 ssl;

```

```

ssl_certificate /etc/nginx/certs/nginx.crt;
ssl_certificate_key /etc/nginx/certs/nginxkey.key;
ssl_trusted_certificate /etc/nginx/certs/ca-certs.crt;
location /
{
    proxy_pass https://wa_server_backend_https;
    proxy_cache off;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header X-Forwarded-Host $host;
    proxy_set_header X-Real-IP $remote_addr;

    proxy_set_header Connection "close";
}
}
}

```

where:

DWCx_HOSTNAME:DWCx_PORT

is the address of the Dynamic Workload Console.

MDMx_HOSTNAME:MDMx_PORT

is the address of the master domain manager.

IBM Workload Scheduler console messages and prompts

The IBM Workload Scheduler control processes (Netman, Mailman, Batchman, Jobman, and Writer) write their status messages (referred to as console messages) to standard list files. These messages include the prompts used as job and Job Scheduler dependencies. On UNIX® and Linux® operating systems, the messages can also be directed to the **syslog** daemon (**syslogd**) and to a terminal running the IBM Workload Scheduler console manager. These features are described in the following sections.

Setting sysloglocal on UNIX™

About this task

If you set **sysloglocal** in the local options file to a positive number, IBM Workload Scheduler's control processes send their console and prompt messages to the **syslog** daemon. Setting it to **-1** turns this feature off. If you set it to a positive number to enable system logging, you must also set the local option **stdlistwidth** to **0**, or a negative number.

IBM Workload Scheduler's console messages correspond to the following **syslog** levels:

LOG_ERR

Error messages such as control process abends and file system errors.

LOG_WARNING

Warning messages such as link errors and stuck job streams.

LOG_NOTICE

Special messages such as prompts and tellops.

LOG_INFO

Informative messages such as job launches and job and Job Scheduler state changes.

Setting **sysloglocal** to a positive number defines the syslog facility used by IBM Workload Scheduler. For example, specifying **4** tells IBM Workload Scheduler to use the local facility LOCAL4. After doing this, you must make the appropriate entries in the **/etc/syslog.conf** file, and reconfigure the syslog daemon. To use LOCAL4 and have the IBM Workload Scheduler messages sent to the system console, enter the following line in **/etc/syslog.conf**:

```
local4    /dev/console
```

To have the IBM Workload Scheduler error messages sent to the **maestro** and **root** users, enter the following command:

```
local4.err  maestro,root
```

The selector and action fields must be separated by at least one tab. After modifying **/etc/syslog.conf**, you can configure the **syslog** daemon by entering the following command:

```
kill -HUP `cat /etc/syslog.pid`
```

console command

About this task

You can use the conman **console** command to set the IBM Workload Scheduler message level and to direct the messages to your terminal. The message level setting affects only Batchman and Mailman messages, which are the most numerous. It also sets the level of messages written to the standard list file or files and the **syslog** daemon. The following command, for example, sets the level of Batchman and Mailman messages to **2** and sends the messages to your computer:

```
console sess;level=2
```

Messages are sent to your computer until you either run another **console** command, or exit conman. To stop sending messages to your terminal, enter the following conman command:

```
console sys
```

Modifying jobmon service rights for Windows™

On Windows™ systems, the IBM Workload Scheduler jobmon service runs in the SYSTEM account with the right **Allow Service to Interact with Desktop** granted to it. You can remove this right for security reasons. However, if you do so, it prevents the service from launching interactive jobs that run in a window on the user's desktop. These jobs will be run, but are not accessible from the desktop or from IBM Workload Scheduler and do not have access to desktop resources. As a result, they may run forever or abend due to lack of resources.

Chapter 2. Configuring the Dynamic Workload Console

This chapter describes how to configure Dynamic Workload Console. It is divided into the following sections:

- [Launching in context with the Dynamic Workload Console on page 127](#)
- [Configuring access to the Dynamic Workload Console on page 136](#)
- [Configuring the Dynamic Workload Console for Single Sign-On on page 140](#)
- [Configuring Dynamic Workload Console to use SSL on page 145](#)
- [Customizing your global settings on page 145](#)
- [Disable the What-if Analysis on page 165](#)
- [Configuring High Availability on page 165](#)
- [Configuring Dynamic Workload Console to view reports on page 166](#)

Launching in context with the Dynamic Workload Console

Create a URL to launch the Dynamic Workload Console and have it directly open the results of a particular query.

By accessing the bookmark icon in the page of your interest, you can copy the URL of that page, and then include the copied URL in an external application, for example, to monitor jobs and job streams that are critical to your business, and to quickly and easily manage them.

Open your Dynamic Workload Console pages and access the information you need in just one click.

Launch in context: your environment at your fingertips.

Scenarios

The following main scenarios can be identified:

- Obtain the result of a monitor query on:
 - Jobs
 - Critical jobs
 - Job streams
 - Workstation
 - Workload
 - Existing tasks

For all the scenarios, you must create a basic URL.

Creating a basic URL

About this task

To create a basic URL, you need to define the URL to access the Dynamic Workload Console:

```
https://{WebUIHostName:adminSecurePort}  
/console/?pageId=<pageID>
```

where:

WebUIHostname

It is the fully qualified hostname or the IP address of the computer where the Dynamic Workload Console is installed.

adminSecurePort

It is the number of the port on which the Dynamic Workload Console is listening.

pageId

It is the ID of the Dynamic Workload Console page that has to be launched.

Example

```
https://mypc:29443/console/?pageId=manage-roles
```

Advanced optional parameters

Depending on the query whose results you want to view, you can complete your URL with the following parameters:

Mandatory parameters

engineName

Specify the name of one or more engines to be used as filter.

objectType

Specify an object type as filter. The following are the supported object types in the Dynamic Workload Console:

- com.ibm.tws.objects.plan.JobInPlan
- com.ibm.tws.objects.plan.JobStreamInPlan
- com.ibm.tws.objects.plan.CriticalJobInPlan
- com.ibm.tws.objects.plan.WorkstationInPlan
- com.ibm.tws.objects.plan.PromptInPlan
- com.ibm.tws.objects.plan.ResourceInPlan
- com.ibm.tws.objects.plan.DomainInPlan

plan

Specify a plan name as filter.

query

Specify a query to filter the results.



Note: Special characters must be replaced by encoded values. For example, # (number sign) must be replaced by %23, + (plus sign) must be replaced by %2B.

**Example**

A query like `!@.#+state=#Waiting` must be written like this: `!@.#+state=#Waiting`

Optional parameters**columns**

Specify the columns that you want to display in your result table. The following three options are available:

ALL

Display all columns.

DEFAULT

Display only the default columns.

Customized columns

Display the customized column to get a specific column result. For example,

```
"columns": "Status, Internal Status".
```

If not specified, the default columns for this query are shown.

encrypt

Specify if you want to encrypt or not the engine name parameter. If true, the engine name has to be encrypted by using Base64 encode.

taskName

Specify the name of an existing task as filter.



Note: it is not recommended to specify both the taskName and query parameters in the URL. If both parameters are specified, the taskName parameter has the priority.

Monitor Jobs on distributed systems

The following is an example of URL to be launched to directly open the Monitor Workload page and obtain results about jobs on distributed systems.

To create a URL to monitor jobs on a distributed system, specify the following filters:

engineName

Specify the name of one or more engines to be used as filter.

objectType

Specify an object type as filter. The following are the supported object types in the Dynamic Workload Console:

- com.ibm.tws.objects.plan.JobInPlan
- com.ibm.tws.objects.plan.JobStreamInPlan
- com.ibm.tws.objects.plan.CriticalJobInPlan
- com.ibm.tws.objects.plan.WorkstationInPlan
- com.ibm.tws.objects.plan.PromptInPlan
- com.ibm.tws.objects.plan.ResourceInPlan
- com.ibm.tws.objects.plan.DomainInPlan

plan

Specify a plan name as filter.

query

Specify a query to filter the results.



Note: Special characters must be replaced by encoded values. For example, # (number sign) must be replaced by %23, + (plus sign) must be replaced by %2B.

Example

A query like `@!@.@+state=#Waiting` must be written like this: `@!@.@%2Bstate=%23Waiting`

columns

Specify the columns that you want to display in your result table. The following three options are available:

ALL

Display all columns.

DEFAULT

Display only the default columns.

Customized columns

Display the customized column to get a specific column result. For example, `"columns": "Status, Internal Status"`.

If not specified, the default columns for this query are shown.

encrypt

Specify if you want to encrypt or not the engine name parameter. If true, the engine name has to be encrypted by using Base64 encode.

Example:

```
https://mypc:9449/console?pageId=direct-query&properties={"query": "%2F%40%2F%40%23%2F%40%2F%40.%40", "engineName": "eJxzzUvPzEs1BAAKagKI", "encrypt": true, "plan": "current-plan", "objectType": "com.ibm.tws.objects.plan.JobInPlan", "columns": "Status, Internal Status, Folder (Job Stream), Job, Job Type, Workstation (Job), Job Stream, Workstation (Job Stream), Scheduled Time, Not Satisfied Dependencies, Priority, Job number, Earliest Start, Actual Start, Deadline"}
```

Monitor Jobs on z/OS® systems

The following is an example of URL to be launched to directly open the Monitor Workload page and obtain results about jobs on z/OS® systems.

To create a URL to monitor jobs on a z/OS® system, specify the following filters:

engineName

Specify the name of one or more engines to be used as filter.

objectType

Specify an object type as filter. The following are the supported object types in the Dynamic Workload Console:

- com.ibm.tws.objects.plan.JobInPlan
- com.ibm.tws.objects.plan.JobStreamInPlan
- com.ibm.tws.objects.plan.CriticalJobInPlan
- com.ibm.tws.objects.plan.WorkstationInPlan
- com.ibm.tws.objects.plan.PromptInPlan
- com.ibm.tws.objects.plan.ResourceInPlan
- com.ibm.tws.objects.plan.DomainInPlan

plan

Specify a plan name as filter.

query

Specify a query to filter the results.



Note: Special characters must be replaced by encoded values. For example, # (number sign) must be replaced by %23, + (plus sign) must be replaced by %2B.

Example

A query like `@!@.@+state=#Waiting` must be written like this: `@!@.@%2Bstate=%23Waiting`

columns

Specify the columns that you want to display in your result table. The following three options are available:

ALL

Display all columns.

DEFAULT

Display only the default columns.

Customized columns

Display the customized column to get a specific column result. For example, `"columns": "Status, Internal Status"`.

If not specified, the default columns for this query are shown.

encrypt

Specify if you want to encrypt or not the engine name parameter. If true, the engine name has to be encrypted by using Base64 encode.

Example:

```
https://
mypc:9449/console?pageId=direct-query&properties={"query":"%40!%40","engineName":"eJyLyI92zUvPzEsFABGsA5M=","e
ncrypt":true,"plan":"current plan","objectType":"com.ibm.tws.objects.plan.JobInPlan","columns":"Status,Internal
Status,Job Number,Job,Workstation,Job stream,Status Details,Scheduled Time,Job Identifier,Error Code,Time
Dependent,Earliest Start,Planned Start,Actual Start,Deadline,Critical"}
```

Monitor Critical Jobs

The following is an example of URL to be launched to directly open the Monitor Workload page and obtain results about critical jobs.

To create a URL to monitor critical jobs, specify the following filters:

engineName

Specify the name of one or more engines to be used as filter.

objectType

Specify an object type as filter. The following are the supported object types in the Dynamic Workload Console:

- com.ibm.tws.objects.plan.JobInPlan
- com.ibm.tws.objects.plan.JobStreamInPlan
- com.ibm.tws.objects.plan.CriticalJobInPlan
- com.ibm.tws.objects.plan.WorkstationInPlan
- com.ibm.tws.objects.plan.PromptInPlan
- com.ibm.tws.objects.plan.ResourceInPlan
- com.ibm.tws.objects.plan.DomainInPlan

plan

Specify a plan name as filter.

query

Specify a query to filter the results.



Note: Special characters must be replaced by encoded values. For example, # (number sign) must be replaced by %23, + (plus sign) must be replaced by %2B.

**Example**

A query like `@!@.@+state=#Waiting` must be written like this: `@!@.@%2Bstate=%23Waiting`

columns

Specify the columns that you want to display in your result table. The following three options are available:

ALL

Display all columns.

DEFAULT

Display only the default columns.

Customized columns

Display the customized column to get a specific column result. For example, `"columns":`

```
"Status, Internal Status".
```

If not specified, the default columns for this query are shown.

encrypt

Specify if you want to encrypt or not the engine name parameter. If true, the engine name has to be encrypted by using Base64 encode.

Example:

```
https://mypc:9449/console?pageId=direct-query&properties={"query":"%2F%40%2F%40%23%2F%40%2F%40.%40", "engineName": "eJxzzUvPzEs1BAAKagKI", "encrypt": true, "plan": "current-plan", "objectType": "com.ibm.tws.objects.plan.CriticalJobInPlan", "columns": "Risk level, Confidence Factor, Status, Internal Status, Folder (Job Stream), Job, Job Type, Workstation (Job), Job Stream, Workstation (Job Stream), Scheduled Time, Jobs Left on Critical Path, Critical Path Remaining Duration, Estimated Start, Estimated End, Earliest Start, Actual Start, Deadline, Critical Latest Start"}
```

Monitor Job Streams

The following is an example of URL to be launched to directly open the Monitor Workload page and obtain results about job streams.

To create a URL to monitor job streams, specify the following filters:

engineName

Specify the name of one or more engines to be used as filter.

objectType

Specify an object type as filter. The following are the supported object types in the Dynamic Workload Console:

- com.ibm.tws.objects.plan.JobInPlan
- com.ibm.tws.objects.plan.JobStreamInPlan
- com.ibm.tws.objects.plan.CriticalJobInPlan
- com.ibm.tws.objects.plan.WorkstationInPlan

- com.ibm.tws.objects.plan.PromptInPlan
- com.ibm.tws.objects.plan.ResourceInPlan
- com.ibm.tws.objects.plan.DomainInPlan

plan

Specify a plan name as filter.

query

Specify a query to filter the results.



Note: Special characters must be replaced by encoded values. For example, # (number sign) must be replaced by %23, + (plus sign) must be replaced by %2B.

Example

A query like `@!@.@+state=#Waiting` must be written like this: `@!@.@%2Bstate=%23Waiting`

columns

Specify the columns that you want to display in your result table. The following three options are available:

ALL

Display all columns.

DEFAULT

Display only the default columns.

Customized columns

Display the customized column to get a specific column result. For example, `"columns":`

```
"Status, Internal Status".
```

If not specified, the default columns for this query are shown.

encrypt

Specify if you want to encrypt or not the engine name parameter. If true, the engine name has to be encrypted by using Base64 encode.

Example:

```
https://mypc:9449/console?pageId=direct-query&properties={"query": "%2F%40%2F%40%23%2F%40%2F%40", "engineName": "eJxzzUvPzEs1BAAKagKI", "encrypt": true, "plan": "current-plan", "objectType": "com.ibm.tws.objects.plan.JobStreamInPlan", "columns": "Status, Internal Status, Folder, Job Stream, Workstation, Scheduled Time, Not Satisfied Dependencies, Total Jobs, Successful Jobs, Jobs Limit, Priority, Earliest Start, Actual Start, Deadline"}
```

Monitor Workstations

The following is an example of URL to be launched to directly open the Monitor Workload page and obtain results about workstations.

To create a URL to monitor workstations, specify the following filters:

engineName

Specify the name of one or more engines to be used as filter.

objectType

Specify an object type as filter. The following are the supported object types in the Dynamic Workload Console:

- com.ibm.tws.objects.plan.JobInPlan
- com.ibm.tws.objects.plan.JobStreamInPlan
- com.ibm.tws.objects.plan.CriticalJobInPlan
- com.ibm.tws.objects.plan.WorkstationInPlan
- com.ibm.tws.objects.plan.PromptInPlan
- com.ibm.tws.objects.plan.ResourceInPlan
- com.ibm.tws.objects.plan.DomainInPlan

plan

Specify a plan name as filter.

query

Specify a query to filter the results.



Note: Special characters must be replaced by encoded values. For example, # (number sign) must be replaced by %23, + (plus sign) must be replaced by %2B.

Example

A query like `@!@.@+state=#Waiting` must be written like this: `@!@.@%2Bstate=%23Waiting`

columns

Specify the columns that you want to display in your result table. The following three options are available:

ALL

Display all columns.

DEFAULT

Display only the default columns.

Customized columns

Display the customized column to get a specific column result. For example, `"columns":`
`"Status, Internal Status".`

If not specified, the default columns for this query are shown.

encrypt

Specify if you want to encrypt or not the engine name parameter. If true, the engine name has to be encrypted by using Base64 encode.

Example:

```
https://mypc:9449/console?pageId=direct-query&properties={"query":"%2F%40%2F%40","engineName":"eJxzzUvPzEs1BAAKagKI","encrypt":true,"plan":"current-plan","objectType":"com.ibm.tws.objects.plan.WorkstationInPlan","columns":["Link Status,Folder,Workstation,Agent Running,Writer Running,Start Time,Run Number,Limit,Domain,Type,Version"]}
```

Existing task

The following is an example of URL to be launched to directly open the Monitor Workload page and obtain results about an existing task.

About this task

To create a URL to monitor an existing task, specify the following filters:

taskName

Specify the name of an existing task as filter.



Note: it is not recommended to specify both the taskName and query parameters in the URL. If both parameters are specified, the taskName parameter has the priority.

Example:

```
https://mypc:9449/console?pageId=direct-query&properties={"taskName":"MyTask"}
```

Configuring access to the Dynamic Workload Console

As soon as you finish installing the Dynamic Workload Console, you can launch it by logging in at the following link:

```
https://<your_ip_address>:9443/console/login.jsp
```

You can access the Dynamic Workload Console from any computer in your environment using a web browser through the secure HTTPS protocol and using the credentials specified at installation time.

By default, the Dynamic Workload Console is configured to use a local file-based user repository. Users defined in the user registry can log in to the Dynamic Workload Console and need to be associated to a role to be able access the Dynamic Workload Console features (see [“Configuring roles to access the Dynamic Workload Console on page 137.”](#))

If you use a central user registry that is based on the Lightweight Directory Access Protocol (LDAP) to manage users and groups and provide single sign-on, then you can set up an LDAP server and create an LDAP user registry to use with the Dynamic Workload Console. You can implement an LDAP user repository in place of the default file-based user registry by configuring the sample authentication templates provided in XML format. The following are the supported LDAP servers and the corresponding sample template that can be configured to replace the configuration file currently in use:

- File-based: `auth_basicRegistry_config.xml`
- IBM® Directory Server: `auth_IDS_config.xml`
- OpenLDAP: `auth_OpenLDAP_config.xml`
- Windows™ Server Active Directory: `auth_AD_config.xml`

In addition you can also add a line

See [Configuring IBM Workload Scheduler using templates on page 428](#) for more information about the templates and the location.



Note: If two or more instances of Dynamic Workload Console share the same database repository for their settings, but they are not configured to be in a High Availability configuration, they all must be at the same fix pack level.

Configuring a user registry

To use LDAP user registry, users and groups must be created by the system administrator in the chosen LDAP server database.

Configuring user registries for the Dynamic Workload Console and all other IBM Workload Scheduler components is described in *Configuring LDAP* described in *Planning and Installation Guide*.

Configuring roles to access the Dynamic Workload Console

During the Dynamic Workload Console installation, new predefined roles are created. They determine which console panels are available to a user, and therefore which activities that user can perform from Dynamic Workload Console. More roles can be created and customized according to business needs. To see a guide for the creation of customized roles see: [Customizing roles](#)

Tip

It is not necessary to assign a role to every single user. If the user registry already contains groups of users that are properly defined for using the console, it is possible to assign roles to groups too. If groups are not available in the user registry, then the special role **all authenticated users** can be used to assign roles to *all* the users at once.

To assign roles to a default groups of users that are properly defined for using the console, add this property to the authentication file in use.

```
<jndiEntry jndiName="all.authenticated.users" value="my-group" />
```

Within the Dynamic Workload Console, you can create your own custom views to enable users to see all or a subset of IBM Workload Scheduler pages. To do it, you must have the **Administrator** role and perform the following steps:

1. Open the `authentication_config.xml` located in the following path:

On UNIX operating systems

`DWC_DATA_dir/usr/servers/dwcServer/configDropins/overrides`

On Windows operating systems

`DWC_home\usr\servers\dwcServer\configDropins\overrides`

2. Add the entity specifying username and password in users or groups.
3. Open **Dynamic Workload Console > Administration > Manage Roles**
4. Click `Entities` to associate the user or the group you have created to one of the roles from the list.
5. Add the entity and save.

The following lists the predefined roles created in WebSphere Application Server Liberty Base for accessing the IBM Workload Scheduler environments using Dynamic Workload Console:

API User

Users in this group can use only the Dynamic Workload Console APIs to perform the available actions. Logging to the Dynamic Workload Console through a Web browser would not give them access to any feature. For more details about the Dynamic Workload Console APIs, see `https://<DWC_hostname>:<port>/dwc/api`.

Administrator

Users with this role can see the entire portfolio and use all features of the Dynamic Workload Console.

Users with this group can also access and use all features of the Self-Service Catalog and the Self-Service Dashboards mobile applications. From the Self-Service Catalog mobile application, these users can create and edit catalogs, create and edit services, add services to catalogs, submit services associated to job streams, and share catalogs and services with other users. From the Self-Service Dashboards mobile application, these users can create and edit dashboards to filter for jobs and workstations, display a dashboard of results, perform recovery actions on a single result.

From the Manage Roles panel, the administrator can add entities, manage pinned pages and shared boards.

Analyst

Users in this group can manage Dynamic Workload Console reports and user preferences.

Broker

Users in this group can define Broker settings, create and manage Broker jobs and resources, and monitor Broker computers and resources.

Developer

Users in this group can create, list, and edit workload definitions, workstations, and event rule definitions in the IBM Workload Scheduler database.

Mobile User

Users in this group can manage the Self-Service Catalog and the Self-Service Dashboards mobile applications but the actions they can perform are limited to submitting service requests (job streams) from the Self-Service Catalog and , from the Self-Service Dashboards mobile application, displaying a dashboard of results and performing recovery actions on them.

Operator

Users in this group can see Dynamic Workload Console:

- All Monitor tasks.
- Jobs and job streams to be submitted on request
- Set User Preferences

The following table lists some entries of the navigation toolbar, and some activities that you can perform on the Dynamic Workload Console. Beside each item, the table shows the groups whose users are authorized to access them.

Table 24. Menu and Group Permissions

Menu Item	Groups with Permission
Quick Start	Administrator
All Configured Tasks	Administrator, Operator
Manage Workload Reports	Administrator, Analyst
Design -> Workload Definitions	Administrator Developer
Planning & Submission -> Workload Forecast	Administrator, Operator
Administration -> Workload Submission	Administrator Operator
Monitoring & Reporting	Administrator, Operator
Design -> Workload Definitions	Administrator
Reporting	Administrator, Analyst

Table 24. Menu and Group Permissions (continued)

Menu Item	Groups with Permission
Security ->Manage Engines	Administrator
Security -> Manage Settings	Administrator
Administration → Broker Settings; Design → Broker Design; Monitoring & Reporting → Broker Monitoring	Administrator, Broker

Configuring the Dynamic Workload Console for Single Sign-On

Single Sign-On (SSO) is a method of access control that allows a user to authenticate once and gain access to the resources of multiple applications sharing the same user registry.

This means that using SSO you can run queries on the plan or manage object definitions on the database accessing the engine without authenticating, automatically using the same credentials you used to log in to the Dynamic Workload Console.

The same is true when working with the Self-Service Catalog and Self-Service Dashboards apps from a mobile device. If the Dynamic Workload Console has been configured to use SSO, then these apps automatically use the same credentials used to log in to the Dynamic Workload Console.

After the installation completes, you can configure the Dynamic Workload Console and the IBM Workload Scheduler engine to use SSO. To do this, they must share the same LDAP user registry.

The Lightweight Directory Access Protocol (LDAP) is an application protocol for querying and modifying directory services running over TCP/IP - see the information about configuring a common LDAP for both the master and console in the post-installation section of the *Planning and Installation Guide* for more details.

If you configured Dynamic Workload Console to use Single Sign-On with an engine, then, the following behavior is applied:

If engine connection has the user credentials specified in its definitions

These credentials are used. This behavior regards also engine connections that are shared along with their user credentials.

If the user credentials are not specified in the engine connection

The credentials you specified when logging in to Dynamic Workload Console are used. This behavior regards also shared engine connections having unshared user credentials.

Before you proceed, ensure that the contents of the `ltpa.keys` file are identical on both the Dynamic Workload Console and the master domain manager. The file is located in the following path:

```
usr/servers/engineServers/resources/security
```

For more information about how to verify and correct this setting, see [How to configure the Dynamic Workload Console and the master domain manager for Single Sign-On on page 141](#).

How to configure the Dynamic Workload Console and the master domain manager for Single Sign-On

Configure the Dynamic Workload Console and the master domain manager for Single Sign-On.

About this task



Note: When implementing a configuration in Single Sign-On, ensure you have not specified the engine credentials in the **Manage Engine** section.

To enable Single Sign-On between the Dynamic Workload Console and master domain manager, perform the following steps:

1. Configure the Lightweight Directory Access Protocol (LDAP) for the Dynamic Workload Console as explained in the post-installation section of the *Planning and Installation Guide*.
2. Create the Access Control list for the LDAP group. For example, to give full access on domain and folders to the LDAP group perform the following steps:
 - a. From the Dynamic Workload Console open the **Manage Workload Security** panel and select **Give access to users and groups**.
 - b. Select the LDAP group from the drop-down list and **FULLCONTROL** in the field **Role**.
 - c. Select **Domain** and assign **ALLOBJECTS**.
 - d. **Save and create new**
 - e. Select the LDAP group from the drop-down list and **FULLCONTROL** in the field **Role**.
 - f. Select **Folder** and assign the root by clicking **/**.
 - g. **Save**
3. On the workstation where the master domain manager is installed, copy the template file located in the following directory to a temporary directory:


```
TWA_home/usr/servers/engineServer/configDropins/templates
```
4. Edit the template file with the information about your LDAP server.
5. Make a backup copy of the existing `authentication_config.xml` file located in the following path:

On UNIX operating systems

master domain manager

```
TWA_DATA_DIR/usr/servers/engineServer/configDropins/overrides
```

On Windows operating systems

master domain manager

```
TWA_home\usr\servers\engineServer\configDropins\overrides
```

6. Replace the existing `authentication_config.xml` file with the template you updated with the information about your LDAP server. Ensure the file permissions and ownership are correct.
7. Ensure that the `ltpa.keys` file on both the Dynamic Workload Console and the master domain manager are identical, copying the file from one instance to the other. The file is located as follows:

Dynamic Workload Console

```
DWC_home/usr/servers/dwcServer/resources/security
```

master domain manager

```
TWA_home/usr/servers/engineServer/resources/security
```

8. Restart WebSphere Application Server Liberty Base on both the master domain manager and the Dynamic Workload Console by running `stopAppServer` and `startAppServer`.

How to configure the Dynamic Workload Console 9.5 and a master domain manager 9.4.x for Single Sign-On

How to configure the Dynamic Workload Console 9.5 and a master domain manager 9.4.x for Single Sign-On.

Before you begin

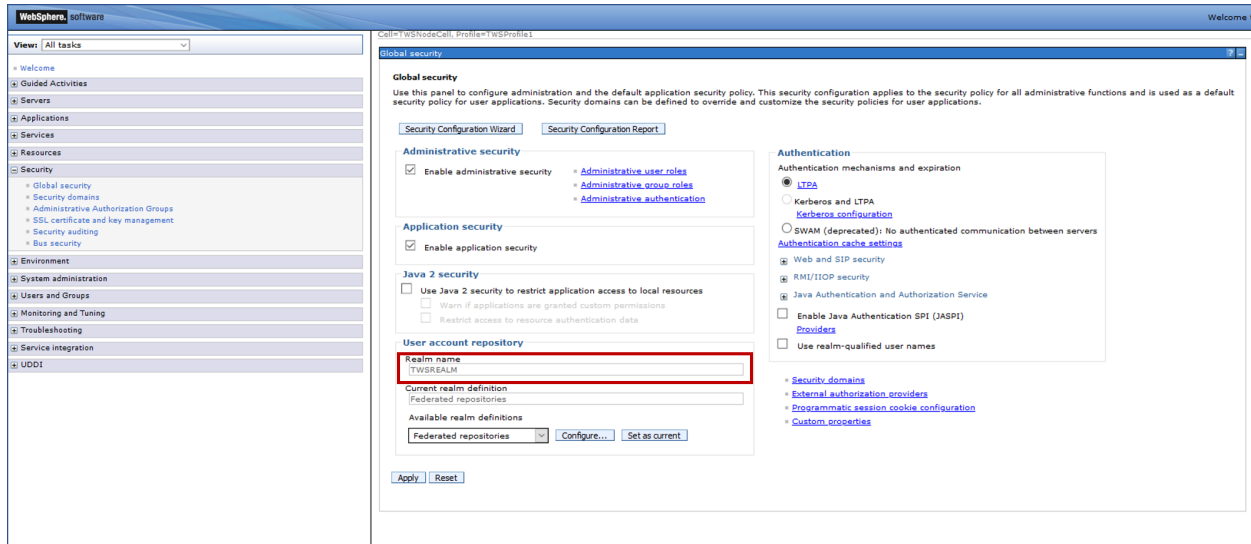
Ensure that the master domain manager V9.4.x is configured to use a Lightweight Directory Access Protocol (LDAP). The LDAP should be the same one already configured and used by the Dynamic Workload Console 9.5. For further information about how to configure an LDAP, see [Configuring LDAP](#).

About this task

To configure the Dynamic Workload Console 9.5 and the master domain manager V9.4.x for Single Sign-On, perform the following steps:

1. Access the **WebSphere administrative console** of the master domain manager V9.4.x and go to **Global security** in the **Security** section.
2. In the Global security panel, take note of the value for the **Realm name** in the *User account repository section*. The realm name is required later in this section.

Figure 2. Realm name in the WebSphere administrative console

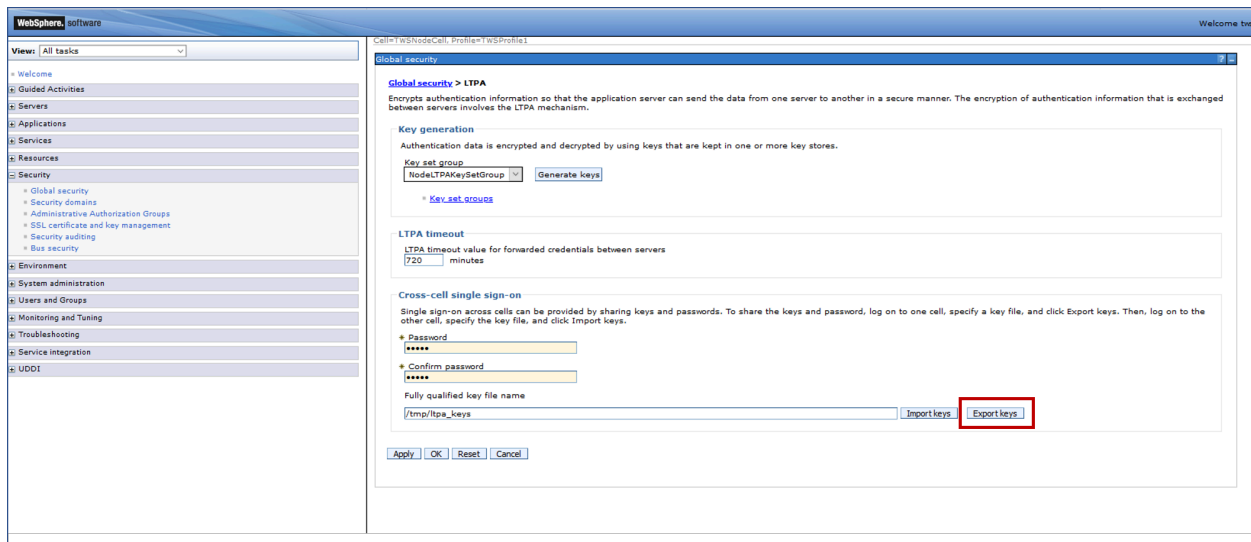


3. In *Authentication*, select **LTPA** as the authentication mechanism, and enter a password to export the Ltpa keys.



Note: Take note of the password. The password you enter is required later during the import.

Figure 3. Export of the Ltpa keys file



4. Before replacing the existing Ltpa on the Dynamic Workload Console 9.5, create a backup copy in a different directory. The existing Ltpa keys file can be found in the following path:

```
DWC_DATA_dir/usr/servers/dwcServer/resources/security/
```

```
DWC_home\usr\servers\dwcServer\resources\security\
```

5. Rename the exported ltpa keys file to **ltpa.keys** and copy it to the same path as the existing file on the Dynamic Workload Console 9.5.
6. Open the authentication configuration file previously customized to enable the LDAP for the Dynamic Workload Console 9.5, and ensure that the realm name is the same as the one specified for the master domain manager V9.4.x (see [Step 2 on page 142](#)). The authentication configuration file is located in the following path:

```
DWC_DATA_dir/usr/servers/dwcServer/configDropins/overrides/
```

```
DWC_home\usr\servers\dwcServer\configDropins\overrides\
```

Figure 4. Realm name in the authentication template

```

59 <federatedRepository searchTimeout="20m">
60 <primaryRealm name="TWSREALM" allowOpIfRepoDown="true">
61 <participatingBaseEntry name="o=BasicRealm"/>
62 <participatingBaseEntry name="{ldap.base.DN}"/>
63 <uniqueGroupIdMapping inputProperty="uniqueName" outputProperty="uniqueName"/>
64 <groupSecurityNameMapping inputProperty="cn" outputProperty="cn"/>
65 <groupDisplayNameMapping inputProperty="cn" outputProperty="cn"/>
66 <userDisplayNameMapping inputProperty="principalName" outputProperty="principalName"/>
67 <userSecurityNameMapping inputProperty="principalName" outputProperty="principalName"/>
68 <uniqueUserIdMapping inputProperty="uniqueName" outputProperty="uniqueName"/>
69 </primaryRealm>
70 </federatedRepository>
71

```

Where *TWSREALM* is the default realm name.

7. Add the password in XOR format in the **ssl_config.xml** as follows:
 - a. Copy the **ssl_config.xml** file from the following path:

```
DWC_home/usr/servers/dwcServer/configDropins/defaults/
```

```
DWC_home\usr\servers\dwcServer\configDropins\defaults\
```

- b. Paste the **ssl_config.xml** file in the following path:

```
DWC_DATA_dir/usr/servers/dwcServer/configDropins/overrides/
```

```
DWC_home\usr\servers\dwcServer\configDropins\overrides\
```


- c. Open the `ssl_config.xml` file and enter the password in XOR format. The password is the one you specified for the master domain manager V9.4.x during the export (see [Step 3 on page 143](#)).

Figure 5. Password in XOR format

```

1 <server description="sslSettings">
2
3
4 <jndiEntry id="keyStore.location" jndiName="keyStore.location" decode="false" value="{keyStore.location}"/>
5 <jndiEntry id="keyStore.password" jndiName="keyStore.password" decode="false" value="{keyStore.password}"/>
6 <jndiEntry id="trustStore.location" jndiName="trustStore.location" decode="false" value="{trustStore.location}"/>
7 <jndiEntry id="trustStore.password" jndiName="trustStore.password" decode="false" value="{trustStore.password}"/>
8
9
10 <keyStore id="twaKeyStore" location="{keyStore.location}" password="{keyStore.password}" type="{keyStore.type}" pollingRate="5s" updateTrigger="{keyStore.trigger}" />
11 <keyStore id="twaTrustStore" location="{trustStore.location}" password="{trustStore.password}" type="{trustStore.type}" pollingRate="5s" updateTrigger="{trustStore.trigger}" />
12 <ssl id="twaSSLSettings" keyStoreRef="twaKeyStore" trustStoreRef="twaTrustStore" sslProtocol="TLSv1.2" clientAuthenticationSupported="true"/>
13 <sslDefault sslRef="twaSSLSettings"/>
14 <ltpa keysPassword="{xor}0so5FiozKW==" keysFileName="{server.config.dir}/resources/security/ltpa.keys" expiration="1440"/>
15 <webAppSecurity ssoUseDomainFromURL="false"/>
16 <httpSession invalidationTimeout="5h" invalidateOnUnauthorizedSessionRequestException="true"/>
17 </server>
18

```

8. Restart the Dynamic Workload Console 9.5.

Results

You successfully configured the Dynamic Workload Console 9.5 and the master domain manager V9.4.x for Single Sign-On.

Configuring Dynamic Workload Console to use SSL

The Secure Sockets Layer (SSL) protocol provides secure communications between remote server processes or applications. SSL security can be used to establish communications inbound to, and outbound from, an application. To establish secure communications, a certificate, and an SSL configuration must be specified for the application.

Full details are supplied in the scenario about the connection between the Dynamic Workload Console and the IBM® Workload Scheduler component that has a distributed connector in *Planning and Installation Guide*.

Customizing your global settings

How to customize global settings.

About this task

To customize the behavior of the Dynamic Workload Console, you can optionally configure some advanced settings. These settings are specified in a customizable file named `TdwcGlobalSettings.xml.template`.

By default, the customizable file is copied into the following path after you install the Dynamic Workload Console:

On Windows operating systems:

```
DWC_home\usr\servers\dwcServer\registry\TdwcGlobalSettings.xml.template
```

On UNIX and Linux operating systems:

```
DWC_home/usr/servers/dwcServer/configDropins/templates/
TdwcGlobalSettings.xml.template
```

If you have Administrator privileges, you can modify the file to replace default values with customized ones and enable commented sections. To enable commented sections, remove the tags that enclose the section. You then save the file locally with the name `TdwcGlobalSettings.xml`.



Note: On UNIX and Linux operating systems, you need to save the `TdwcGlobalSettings.xml` file in the following path: `DWC_DATA_dir/usr/servers/dwcServer/registry/TdwcGlobalSettings.xml`

You can add and modify some customizable information, such as:

- The URLs that link to videos in the Dynamic Workload Console. For example, you can link to a company intranet server to view help videos rather than to a public video site.
- The maximum number of objects to be shown in the graphical views.
- The setting to display the plan view in a new window.
- The auto refresh interval for the **Show Plan View** graphical view.
- The creation of predefined tasks.
- The URLs where you can store customized documentation about your jobs or job streams to associate customized documentation to them.
- The current user registry in use.
- The timeout to read and write information on a IBM Z Workload Scheduler engine.
- The maximum number of objects to be retrieved with a query, the maximum number of rows to display in a table, and the maximum number of direct queries to maintain in history.
- Allowing or preventing users from sharing tasks and engine connections.
- The display of all dependencies, both satisfied and unsatisfied.
- The use of audit files to track activities in the Self-Service Catalog and Self-Service Dashboards mobile applications.
- Displaying or hiding all predecessors from the What-if Analysis Gantt view.

This file is accessed at each login, and all configurations specified in the file are immediately applied, except for the **precannedTaskCreation** property. This property is read only when a user logs in for the first time and is then used whenever this user logs in again.

You can use any text or XML editor to edit this file, but ensure that you save it as a valid XML file.

The file is organized into sections that group similar properties. An explanation of each section is available in the file. For more information, see [TdwcGlobalSettings.xml sample on page 160](#).

Sections can also be repeated multiple times in the same file and applied differently to different user roles. To apply a section only to the users belonging to a role, the section must be included within the tags `<settings role="user_role">` and `</settings>`, where:

<user_role>

The user for which the enclosed configuration must be applied. The default value is all users, unless otherwise specified.

Only one **settings** section can be specified for each role. If a user has more than one role, the settings associated to the higher role are used.

To edit the file, proceed as follows:

1. Stop WebSphere Application Server Liberty Base using the following command:

UNIX™

Stop the application server

```
./stopAppServer.sh [-direct]
```

Windows™

Stop the application server

```
stopAppServer.bat [-direct
                  [-wlpHome <installation_directory>]
                  [-options <parameters>]]
```

as described in the section about starting and stopping WebSphere Application Server Liberty Base in *Administration Guide*.

2. Log in as root or Administrator to the Dynamic Workload Console.
3. Browse to

On Windows operating systems:

`DWC_home\usr\servers\dwcServer\registry\TdwGlobalSettings.xml.template`

On UNIX and Linux operating systems:

`DWC_home/usr/servers/dwcServer/configDropins/templates/
TdwGlobalSettings.xml.template`

4. Edit the file as necessary, rename it to `TdwGlobalSettings.xml` and save it.



Note: On UNIX and Linux operating systems, you need to save the `TdwGlobalSettings.xml` file in the following path: `DWC_DATA_dir/usr/servers/dwcServer/registry/TdwGlobalSettings.xml`

5. Start WebSphere Application Server Liberty Base using the following command:

UNIX™

Start the application server

```
./startAppServer.sh [-direct]
```

Windows™

Start the application server

```
startAppServer.bat [-direct]
```

as described in the section about starting and stopping WebSphere Application Server Liberty Base in *Administration Guide*.

Example:

```
<?xml version="1.0"?>
<tdwc>
.
.
<settings>
<graphViews>
<property name="planViewNewWindow" value="true"/>
</graphViews>
</settings>

<settings role="TWSWEBUIOperator">
<graphViews>
<property name="planViewNewWindow" value="false"/>
</graphViews>
</settings>
.
.
</tdwc>
```

To view the complete syntax for the file, see [TdwcGlobalSettings.xml sample on page 160](#).

Customize video URLs

This section shows how you should customize your URLs that link video content in the Dynamic Workload Console so that you can link to a company intranet server to view help videos rather than a public video site.

The `_baseUrl` prefix will be added to all your video URLs . If you do not specify a link for your video the default setting will automatically be used.

```
<?xml version="1.0"?>
<tdwc>
.
.
<settings>
-<videoGallery>
<property name="_baseUrl" value=""></property>
<property name="depLoop" value=""></property>
<property name="highlightRelDep" value=""></property>
<property name="viewDepPrompt" value=""></property>
<property name="usingImpactView" value=""></property>
<property name="createUseTasks" value=""></property>
<property name="weAddRemoveFile" value=""></property>
<property name="weCreateDeps" value=""></property>
<property name="weAddJob" value=""></property>
```

```
<property name="weHighlightDeps" value=""></property>
<property name="weCreateJCL" value=""></property>
</videoGallery>
```

Override graphical view limits

This section contains the configuration parameters that apply to the graphical views in the plan, such as the maximum number of objects shown in each view.

planViewMaxJobstreams

The maximum number of job streams displayed in the Plan View. Default value is **1000**. Values greater than **1000** are not supported.

preProdPlanViewMaxJobstreams

The maximum number of job streams displayed in the preproduction plan view. Default value is **1000**. Values greater than **1000** are not supported.

```
<?xml version="1.0"?>
<tdwc>
.
.
  <settings>
<graphViews>
<property name="planViewMaxJobstreams" value="1000"></property>
<property name="preProdPlanViewMaxJobstreams" value="1000"></property>
</graphViews>
</settings>
.
.
</tdwc>
```

See [TdwcGlobalSettings.xml sample on page 160](#) to view the complete syntax for the file.

For more information about how to customize global settings, see [Customizing your global settings on page 145](#).

Plan View in new window

This section is used to prevent Internet Explorer 7 from freezing while using the Plan View. To solve the problem, set value to **true**.

planViewNewWindow

Set it to **true** if you want the plan view to be displayed in a new window each time it is launched. Default value is **false**.

```
<?xml version="1.0"?>
<tdwc>
.
.
  <settings>
<graphViews>
<property name="planViewNewWindow" value="true"/>
```

```

</graphViews>
.
.
</settings>
</tdwc>

```

See [TdwcGlobalSettings.xml sample on page 160](#) to view the complete syntax for the file.

For more information about how to customize global settings, see [Customizing your global settings on page 145](#).

Plan View auto refresh interval

Use this section to change the default setting of the auto refresh interval for the Show Plan View graphical view for all users.

By default, the auto refresh interval is 300 seconds (five minutes).

PlanViewAutorefresh

The graphical representation of the Plan View is automatically refresh every 300 seconds by default. To change this setting, edit the value assigned to the **DefaultTime** property. The minimum value you can set is 30 seconds. Any value specified below this value is reset to 30 seconds. You must restart the Dynamic Workload Console application server after modifying this value.

```

<?xml version="1.0"?>
<tdwc>
.
.
  <settings>
<PlanViewAutorefresh>
<property name="DefaultTime" value="300"/>
</PlanViewAutorefresh>
.
.
  </settings>
</tdwc>

```

See [TdwcGlobalSettings.xml sample on page 160](#) to view the complete syntax for the file.

For more information about how to customize global settings, see [Customizing your global settings on page 145](#).

Disable and customize NewsFeed function

This section contains the configuration details to be constantly up-to-date with product information.

FeedURL

Contains the URL from which you receive news and updates. Default value is: <https://www.ibm.com/developerworks/community/wikis/form/anonymous/api/wiki/585f5525-a7f5-48ef-9222-50ad582e85f4/page/e599dd3c-8dc3-4ab6-89fd-33f81a994799/attachment/de677e63-5a9d-46db-a010-18ca38f05812/media/tws.jsonp>

FeedType

A string that identifies the format of update information. Default value is **JSONP**.

PollInterval

The interval in seconds between two checks for updates. Default value is **600**.

PollInitialDelay

An initial delay in seconds before the first attempt to read the news feeds. After the initial load, the poll interval is used. Default value is **120**.

NewsFeed

Property used to add further customized news feeds. Specify the format and address of the file that contains the customized communication. Supported formats are RSS 2.0 and ATOM 1.0. You must write the communication in ATOM 1.0 or RSS 2.0 format and store this file in the an HTTP server complying with the *same origin policy*. For browser security reasons, this policy permits to access information only on server using the same protocol, hostname and port number as the one to which you are connected. Optionally, if you want to store your customized feed on an external server, you must configure an HTTP reverse proxy server mapping the external server address.

```
<property name="NewsFeed" type="RSS"
value="http://DWC_hostname:portnumber.com/news.rss" />
```



Note: To specify multiple feeds, you must specify multiple **NewsFeed** properties.

NewsFeedCategory

The name of the customized information. It can be used to identify informational, warning or alert messages, for example. The path to an image can also be added to better identify the information with an icon.

To add more category images, specify a list of properties named **NewsFeedCategory**, for example:

```
<property name="NewsFeedCategory" value="my company info"
icon="http://www.my.company.com/info.png" />
<property name="NewsFeedCategory" value="my company alert"
icon="http://www.my.company.com/alert.png" />
```

If no customized feed is specified, the default feed is used, which retrieves the latest product information from official support sites. To disable any notification, comment the entire section. To disable only external notifications about product information updates, assign an empty string as value to the `FeedURL` property of `JSONP` feed like:

```
<property name="FeedURL" type="JSONP" value="" />
```

Example:

```
<?xml version="1.0"?>
<tdwc>
.
.
  <settings>
  <NewsFeed>
  <property name="NewsFeed" type="RSS"
value="http://www.DWC_hostname:portnumber.com/my_rss.xml" />
  <property name="NewsFeed" type="ATOM"
value="http://www.DWC_hostname:portnumber.com/my_atom.xml" />
```

```

<property name="PollInterval" value="600" />
<property name="PollInitialDelay" value="1" />

<property name="FeedURL" type="JSONP" value="" />

<property name="NewsFeedCategory"
value="my company info" icon="http://www.DWC_hostname:portnumber.com
/info.png" />
<property name="NewsFeedCategory"
value="my company alert" icon="http://www.DWC_hostname:portnumber.com
/alert.png" />

</NewsFeed>
</settings>
.
.
</tdwc>

```

See [TdwcGlobalSettings.xml sample on page 160](#) to view the complete syntax for the file.

For more information about how to customize global settings, see [Customizing your global settings on page 145](#).

Disable and customize the creation of predefined tasks

This section defines the environment for which predefined tasks are created.

precannedTaskCreation

Some predefined tasks are created by default and are available when you log in to the console. There is a predefined Monitor task for every object, for both z/OS® and distributed engines. Default value is **all**. To change this setting, use one of the following values:

all

All predefined tasks are created. This is the default.

distributed

Only predefined tasks for distributed engines are created

zos

Only predefined tasks for z/OS engines are created

none

No predefined task is created.

```

<?xml version="1.0"?>
<tdwc>
.
.
  <settings>
    <application>
      <property name="precannedTaskCreation" value="all"/>
    </application>
  </settings>
.

```



```
.
</tdwc>
```

See [TdwcGlobalSettings.xml sample on page 160](#) to view the complete syntax for the file.

For more information about how to customize global settings, see [Customizing your global settings on page 145](#).

Add customized URL to job and job streams

This section contains URLs where you can store customized documentation about your jobs or job streams. By default, this setting is not specified. If you want to associate customized documentation to a job or job stream, use this setting to specify the external address where this information is located.

If you want to specify a URL where customized documentation for a job and job stream is stored, uncomment the section lines, specify the required URL, and optionally assign a name to the UI label by specifying a value for the `customActionLabel` property. By default this name is **Open Documentation**. This label is then displayed in the **More Actions** menus in Monitor Jobs and Monitor Job Streams tasks, as well as in the graphical views of the plan (in the object's tooltips, context menus and properties). In this example, selecting **Open Documentation** accesses the relevant documentation making it possible to open the documentation while monitoring your job or job stream in the plan.

To implement this setting, assign values to the following keywords:

customActionLabel

The name of the action displayed in menus, object properties, and tooltips to access customized documentation about your jobs or job streams. By default this name is "Open Documentation" unless you customize the name with this keyword.

jobUrlTemplate

The address of your job documentation. No default value available.

jobstreamUrlTemplate

The address of your job stream documentation. No default value available.

Consider the following example:

```
<?xml version="1.0"?>
<tdwc>
.
.
<settings>
  <twsObjectDoc>
    <property name="jobstreamUrlTemplate"
      value="http://www.yourhost.com/tws/docs/${js_encoded_folder_path}${js_name_w}"/>
    <property name="jobUrlTemplate"
      value="http://www.yourhost.com/docs/jobs/${job_name_w}"/>
    <property name="customActionLabel" value="Your Custom Label Name"/>
  </twsObjectDoc>
</settings>
.
.
</tdwc>
```

See [TdwGlobalSettings.xml sample on page 160](#) to view the complete syntax for the file.

These properties must be valid URLs, containing one or more of the variables listed in the table below.

If you use any of the following special characters in the URL, you must write them as follows:

Table 25. Syntax for special characters

Special characters	Write them as...
<i>quote</i> (")	\"
<i>apostrophe</i> (')	'
<i>ampersand</i> (&)	&
<i>less than</i> (<)	<
<i>greater than</i> (>)	>
<i>backslash</i> (\)	\\

Multiple variables can be included in a URL and must be specified using the following syntax: `${variable}`:

Table 26. Variables used in the URL definition

Name	Object	Description
job_number_w	Job z/OS®	The number of the job
job_wkst_w	Job	The name of the workstation on which the job runs and the folder where it is stored, if any.
job_jsname_w	Job	The name of the job stream that contains the job and the folder where it is stored, if any.
job_jswkst_w	Job	The name of the job stream that contains the job and the folder where it is stored, if any.
job_actualarriva l_w	Job z/OS®	The actual start time of the job (date format: YYYY-MM-DDThh:mm:ss)
job_actualend_w	Job z/OS®	When the job actually completed (date format: YYYY-MM-DDThh:mm:ss)
job_starttime_w	Job	The start time of the job (date format: YYYY-MM-DDThh:mm:ss)
job_id_w	Job	The ID of the job
job_returncode_w	Job	The return code of the job
js_name_w	Job stream	The name of the job stream that contains the job
js_wkst_w	Job stream	The name of the job stream that contains the job and the folder where it is stored, if any.
js_id_w	Job stream	The job stream ID

Table 26. Variables used in the URL definition (continued)

Name	Object	Description
js_latest_start_w	Job stream	The latest time at which a job stream can start (date format: YYYY-MM-DDThh:mm:ss)
engine_name_w	Engine	The name of the engine connection
engine_host_w	Engine	The hostname of the engine connection
engine_port_w	Engine	The port number of the engine connection
engine_plan_w	Engine	The ID of selected plan
engine_serv_w	Engine	The remote server name of the engine connection

User registry

Use this section to configure some properties related to the User Registry in use.

groupIdMap

The property groupIdMap is related to the groups of User Registry, and can be modified to map and display the specified value of each group. By default the common name of the group is displayed.

importSettingsMaxFileSize

The property importSettingsMaxFileSize is related to the "Manage settings" > "Import Settings" functionality and defines the max file size of the uploaded TDWCSettings.xml. KB is the unit of measure, and by default, it is set to 102400 KB (100 MB). If you need to upload a property file bigger than 100MB, you can increase this value, but for security purposes, it is strongly suggested to revert the file size back to the default value once the import has been performed.

Examples:

```
<?xml version="1.0"?>
<tdwc>
.
.
<settings>
<security>
<property name="groupIdMap" value="cn"></property>
<property name="importSettingsMaxFileSize" value="102400"></property>
</security>
</settings>
.
.
</tdwc>
```

Therefore, if you need to change the default value "cn" to "racfid", you can define this property as follows:

```
<property name="groupIdMap" value="racfid"></property>
```

See [xref href="awsadtdwcfglobsetxmp.dita"/>](awsadtdwcfglobsetxmp.dita) to view the complete syntax for the file.

or see User Settings to manage Dynamic Workload Console settings.

For more information about how to customize global settings, see [Customizing your global settings on page 145](#).

z/OS http connections

Use this section to configure the timeout to read and write information on IBM® Z® Workload Scheduler engine. When you connect to the IBM® Z® Workload Scheduler engine to retrieve a list of defined objects, you receive an error message if the list is not returned within the timeout period. The value is expressed in milliseconds.

Example:

```
<?xml version="1.0"?>
<tdwc>
.
.
<settings>
<http>
<property name="zosHttpTimeout" value="90000" />
</http>
.
.
</settings>
</tdwc>
```

See [TdwcGlobalSettings.xml sample on page 160](#) to view the complete syntax for the file.

For more information about how to customize global settings, see [Customizing your global settings on page 145](#).

Limit the number of objects retrieved by queries

Use this section to configure: the number of results displayed for Monitor tasks, the maximum number of rows to display on each page, and the number of direct queries to maintain in history.

If you want to limit the number of results produced by your queries, you can specify the maximum number of items that must be retrieved using the `monitorMaxObjectsPM` property. The minimum number of retrieved results is 500.



Note: `monitorMaxObjectsPM` property only limits the number of results for archived plans queries. The property does not affect current plan queries.

The default value is -1; any value lower than 0 means that there is no limit in the number of objects retrieved.

Because data is extracted in blocks of 250 rows, the value you enter is adjusted to complete an entire block. For example, if you specify a limit of 500, only 500 elements are retrieved, while if you specify a limit of 600, 750 elements are retrieved.

For Multiple engine tasks, this limit is applied to each engine included in the query. Therefore, if you specify a limit of 500 results and, for example, you run a Monitor jobs on multiple engine task on three engines, the results produced by your query will be no more than 500 *for each engine*, for a maximum of 1 500 rows.



Note: This setting does not apply to Monitor critical jobs tasks.

To set the maximum number of rows to display in a table view, configure the `maxRowsToDisplay` property.

To set the maximum number of direct queries to maintain in history, configure the `maxHistoryCount` property. These queries are available from the pull-down for the Query field on the Monitor Workload page.

```
<?xml version="1.0"?>
<tdwc>
.
.
<settings>
  <monitor>
    <property name="monitorMaxObjectsPM" value="2000"></property>
  </monitor>

  <ph rev="v92"><monitor>
    <property name="maxRowsToDisplay" value="25"></property>
  </monitor>

  <monitor>
    <property name="maxHistoryCount" value="100"></property>
  </monitor>
</ph>
</settings>

<settings>
  <search>
    <property name="search_max_limit" value="1500"></property>
  </search>
</settings>
.
.
</tdwc>
```

See [TdwcGlobalSettings.xml sample on page 160](#) to view the complete syntax for the file.

For more information about how to customize global settings, see [Customizing your global settings on page 145](#).

Limit task and engine sharing

Use this section to prevent users from sharing tasks and engines.

By default there is no limit to task and engine sharing and all users are authorized to share their tasks and engine connections. If you want to change this behavior, preventing users from sharing tasks and engines, set this property to **true**.

The property default value is **false**, set it to **true** to enable the limit:

limitShareTask

Set to true to prevent users from sharing tasks.

limitShareEngine

Set to true to prevent users from sharing engine connections.

```

<?xml version="1.0"?>
<tdwc>
.
.
<settings>
  <security>
    <property name="limitShareTask"    value="false" />
    <property name="limitShareEngine"  value="false" />
  </security>
</settings>
.
.
</tdwc>

```

See [TdwcGlobalSettings.xml sample on page 160](#) to view the complete syntax for the file.

For more information about how to customize global settings, see [Customizing your global settings on page 145](#).

Show all dependencies

This section defines whether to show all dependencies displayed, regardless of their being satisfied or not.

ShowDependencies

When you open the dependencies panel from Monitor jobs and Monitor job streams task results, by default only **Not Satisfied** dependencies are shown. Uncomment this section and leave the value set to **"true"** to have all dependencies displayed, regardless of their being satisfied or not. Possible values are:

true

All dependencies displayed, regardless of their being satisfied or not.

false

Only not satisfied dependencies are displayed.

```

<?xml version="1.0"?>
<tdwc>
.
.
<settings>
  <ShowDependencies>
    <property name = "AlwaysShowAllDependencies"
      value="true"></property>
  </ShowDependencies>
</settings>
.
.
</tdwc>

```

See [TdwcGlobalSettings.xml sample on page 160](#) to view the complete syntax for the file.

For more information about how to customize global settings, see [Customizing your global settings on page 145](#).

Auditing mobile app activity

This section defines whether to track activities performed in the Self-Service Catalog and Self-Service Dashboards applications in an auditing log file.

For information about the name and location of the log file, see the logs and traces section in the *Troubleshooting Guide*.

SSAuditing

This value is set to **"true"** by default so that operations performed in the Self-Service Catalog and Self-Service Dashboards applications are written to a log file. The log file contains information such as creation, modification and deletion dates, the operations performed in the mobile apps, and the user performing the operations. Possible values are:

true

Operations performed in the Self-Service Catalog and Self-Service Dashboards applications are tracked in an auditing log file.

false

Operations performed in the Self-Service Catalog and Self-Service Dashboards applications are not tracked in an auditing log file.

SSAuditingLogSize

The maximum size of a log file in KB. When a log file reaches the maximum size, the system rolls that log file over and creates a new file. By default, the maximum size of a log file is 100 KB.

SSAuditingLogFiles

The default number of log files to create. When this number is met and the latest log file reaches its maximum size, the system deletes the oldest log file and rolls the latest file over and creates a new file.

```
<?xml version="1.0"?>
<tdwc>
.
.
<settings>
<SSCAuditing>
    <property name = "SSAuditing"      value="true"></property>
    <property name = "SSAuditingLogSize" value="100"></property>
    <property name = "SSAuditingLogFiles" value="2"></property>
</SSCAuditing>
</settings>
.
.
</tdwc>
```

See [TdwGlobalSettings.xml sample on page 160](#) to view the complete syntax for the file.

For more information about how to customize global settings, see [Customizing your global settings on page 145](#).

Modifying the number of archived plans displayed in the Dynamic Workload Console

You can modify the number of archived plans displayed in the Monitor Workload view of the Dynamic Workload Console. The default number is 30 plans.

To modify the default number, configure the following property in the **TdwcGlobalSettings.xml** file:

```
<monitor>
  <property name="maxArchivedPlan" value="30"></property>
</monitor>
```

See [TdwcGlobalSettings.xml sample on page 160](#) to view the complete syntax for the file.

For more information about how to customize global settings, see [Customizing your global settings on page 145](#).

Show or hide predecessors from What-if Analysis Gantt view

When you have hundreds of predecessors, you can optimize performance by excluding them from the What-if Analysis Gantt view. By default, all predecessors are loaded into the What-if Analysis Gantt view. To exclude them, uncomment this section and leave the default setting of the property **whatIfAutoLoadPreds** to `"false"`. To revert back to the default behavior either set the property to `"true"` or comment the section again in the **TdwcGlobalSettings.xml** file.

To modify the default setting, configure the following property in the **TdwcGlobalSettings.xml** file:

```
<WhatifAnalysis>
  <property name = "whatIfAutoLoadPreds" value="false"></property>
</WhatifAnalysis>
```

See [TdwcGlobalSettings.xml sample on page 160](#) to view the complete syntax for the file.

For more information about how to customize global settings, see [Customizing your global settings on page 145](#).

TdwcGlobalSettings.xml sample

The following example is a sample of the file:

```
<?xml version="1.0"?>
<tdwc>

#####
#####  SETTINGS FOR ALL USERS  #####
#####

<settings>

#####
#####  CUSTOMIZE LINKS TO VIDEOS  #####
#####

This section shows how you should customize your URLs that link video content in
the Dynamic Workload Console so that you can link to a company intranet server
to view help videos rather than a public video site.

### This prefix "_baseUrl" will be added to all video URLs ###

### Links to videos, missing entries or empty (blank) values are not considered ###

#### Graphical view: detect loop          ###
#### Graphical view: highlight and release dependencies ####
```



```

#### Graphical view: reply to prompt dependency      ####
#### Graphical view: using the impact view          ####
#### Table: creating and using tasks                ####
#### Workload editor: add and remove a file dependency ####
#### Workload editor: create a dependency           ####
#### Workload editor: add a job                     ####
#### Workload editor: highlight dependencies        ####
#### Workload editor: creating a z/OS job           ####

<videoGallery>
  <property name="_baseUrl" value=""></property>
  <property name="depLoop" value=""></property>
  <property name="highlightRelDep" value=""></property>
  <property name="viewDepPrompt" value=""></property>
  <property name="usingImpactView" value=""></property>
  <property name="createUseTasks" value=""></property>
  <property name="weAddRemoveFile" value=""></property>
  <property name="weCreateDeps" value=""></property>
  <property name="weAddJob" value=""></property>
  <property name="weHighlightDeps" value=""></property>
  <property name="weCreateJCL" value=""></property>
</videoGallery>

#####
##### SECTION 1 - GRAPHICAL VIEW SETTINGS #####
#####

This section specifies the maximum number of objects shown in each graphical view.
Default value is 1000 for all properties.

<graphViews>
  <property name="planViewMaxJobstreams" value="1000"></property>
  <property name="preProdPlanViewMaxJobstreams" value="1000"></property>
</graphViews>

#####
##### SECTION 2 - PLAN VIEW IN NEW WINDOW #####
#####

This section is used to prevent Internet Explorer 7 from freezing while using the Plan View. To solve the problem, set value      to true.
Default value is false

<graphViews>
  <property name="planViewNewWindow" value="true"/>
</graphViews>

#####
##### SECTION 3 - DISABLE / CUSTOMIZE NEWS FEED FUNCTION #####
#####

This section allows overriding the properties concerning the "NewsFeed" function.
Default values are as follows:
  <NewsFeed>
    <property name="FeedURL"
value="https://www.ibm.com/developerworks/wikis/pages/viewpageattachments.action?pageId=119079645&sortBy=date&highlight=tw.jsonp&" />
    <property name="FeedType" value="JSONP" />
    <property name="PollInterval" value="3600" />
  </NewsFeed>

To disable function

<NewsFeed>
  <property name="FeedURL" value="" />
  <property name="FeedType" value="JSONP" />
  <property name="PollInterval" value="3600" />
</NewsFeed>

Starting from 8.6 FP2 you can allow multiple feeds, and create new categories:
<NewsFeed>
  <property name="NewsFeed" type="RSS" value="https://httpserver.mycompany.com:29443/ibm/TWSWebUI/bulb/info_news_rss.xml" />
  <property name="NewsFeed" type="ATOM" value="https://httpserver.mycompany.com:29443/ibm/TWSWebUI/bulb/alert_news_atom.xml" />
  <property name="PollInterval" value="1" />

```

```

<property name="PollInitialDelay" value="1" />
<property name="FeedURL" type="RSS" value="" />

<property name="NewsFeedCategory" value="InfoNews" icon="https://httpserver.mycompany.com:29443/ibm/TWSWebUI/bulb/info.png" />
<property name="NewsFeedCategory" value="AlertNews" icon="https://httpserver.mycompany.com:29443/ibm/TWSWebUI/bulb/alert.png" />
</NewsFeed>

#####
##### SECTION 4 - DISABLE /CUSTOMIZE CREATION OF PREDEFINED TASKS #####
#####

To avoid or customize the creation of predefined tasks at first logon.
Possible values are:
all          both distributed and z/OS tasks are created. This is the default value
none         no task is created
distributed  only distributed tasks are created
zos          only z/OS tasks are created

<application>
<property name="precannedTaskCreation" value="all"/>
<property name="updateWorkstationMaxNumber" value="20"/>
</application>

<PositionSorting>
<property name="enabled" value="true"></property>
</PositionSorting>

#####
##### SECTION 5 - ADD A CUSTOM DOCUMENTATION URL TO JOB/JOBSTREAM #####
#####

This section contains URLs where you can store customized documentation about your jobs or job streams.
By default this setting is not specified. If you want to associate customized documentation to a job or
job stream, use this setting to specify the external address where this information is located.
If you want to specify a URL to be opened as related documentation for jobs and job streams,
uncomment the section lines so that a new action, Open Documentation, is inserted in the More Actions
menu for Monitor Jobs and Monitor Job Streams tasks. The new action links to the specified URL

You can customize the URL template by using variables. The variables have the following syntax
${<variable_name>}

For the complete list of variables, please refer to the documentation.

<twsojectDoc>
<property name="jobstreamUrlTemplate" value="http://www.yourhost.com/tws/docs/jobstream/${js_name_w}" />
<property name="jobUrlTemplate" value="http://www.yourhost.com/docs/jobs/${job_name_w}" />
<property name="customActionLabel" value="Custom Action" />
</twsojectDoc>

#####
##### SECTION 6 - USER REGISTRY #####
#####

In this section you can configure properties related to the User Registry in use.

The property groupIdMap is related to the groups of User Registry, and can be modified
to map and display the specified value of each group. By default the common name
of the group is displayed.

The property importSettingsMaxFileSize is related to the "Manage settings" > "Import Settings"
functionality and defines the max file size of the uploaded TDWCSsettings.xml.
KB is the unit of measure, and by default, it is set to 102400 KB (100 MB).
If you need to upload a property file bigger than 100MB, you can increase this value, but
for security purposes, it is strongly suggested to revert the file size back to the default
value once the import has been performed.

<security>
  <property name="groupIdMap" value="cn"></property>
  <property name="importSettingsMaxFileSize" value="102400"></property>
</security>

```

```

#####
##### SECTION 7 - Z/OS HTTP CONNECTIONS #####
#####

Use this section to increase or decrease timeout for http connection in Z/OS
environment. Change this setting if you receive a connection timeout
using plugin actions/picklists.

The setting is in milliseconds.

<http>
  <property name="zosHttpTimeout" value="90000" />
</http>

#####
##### SECTION 8 - LIMIT THE NUMBER OF OBJECTS RETURNED BY THE QUERIES #####
#####

Use this section to configure: the number of results displayed for Monitor tasks, the maximum number of rows
to display on each page, and the number of direct queries to maintain in history.
This setting applies to all tasks except for Monitor critical jobs and Monitor jobs on multiple engines.
If you want to limit the number of results produced by your queries, you can specify the maximum number of items that must be retrieved.
The default value is -1; any value lower than 0 means that there is no limit in the number of objects retrieved.
The minimum number of retrieved results is 500. Because data is extracted in blocks of 250 rows,
the value you enter is adjusted to complete an entire block. For example, if you specify a limit of 500,
only 500 elements are retrieved, while if you specify a limit of 600, 750 elements are retrieved.
To set the maximum number of rows to display in a table view, configure the maxRowsToDisplay property.
To set the maximum number of direct queries to maintain in history, configure the maxHistoryCount property.
These queries are available from the pull-down for the Query field on the Direct Query page.

<monitor>
  <property name="monitorMaxObjectsPM" value="2000"></property>
</monitor>
<monitor>
  <property name="maxRowsToDisplay" value="25"></property>
</monitor>

You modify the number of archived plans displayed in the Monitor Workload view of the Dynamic
Workload Console. The default number is 30 plans.

<monitor>
  <property name="maxArchivedPlan" value="30"></property>
</monitor>

<monitor>
  <property name="maxHistoryCount" value="100"></property>
</monitor>

<settings>
<search>
  <property name="search_max_limit" value="500"></property>
</search>
</settings>

#####
##### SECTION 9 - LIMIT TASK AND ENGINE SHARING #####
#####

Use this section to prevent users from sharing tasks and engines.
By default there is no limit to task and engine sharing and all users are authorized to share
their tasks and engine connections. If you want to change this behavior, preventing users from
sharing tasks and engines, set this property to true. The property default value is false,
set it to true to enable the limit:

<security>
  <property name="limitShareTask" value="false" />
  <property name="limitShareEngine" value="false" />
</security>

#####

```

```
##### SECTION 10 - CHANGE DEFAULT BEHAVIOR FOR DEPENDENCIES PANEL #####
#####

Use this section to change the default behavior of the UI when displaying
dependencies in the dependencies panel. By setting this value to true, by default,
all dependencies are displayed, and not just the unsatisfied ones.

<ShowDependencies>
  <property name = "AlwaysShowAllDependencies" value="true"></property>
</ShowDependencies>

##### SECTION 11 - CHANGE DEFAULT BEHAVIOR FOR SSC AND SSD AUDITING #####
#####

Use this section to change the default behavior of the auditing of activities performed
using the Self-Service Catalog and the Self-Service Dashboards application. By default,
auditing is enabled. You can also set the maximum size of the log file before it rolls
over to a new log file, and the maximum number of log files maintained.

<SSCAuditing>
  <property name = "SSAuditing" value="true"></property>
  <property name = "SSAuditingLogSize" value="100"></property>
  <property name = "SSAuditingLogFiles" value="2"></property>
</SSCAuditing>

##### SECTION 12 - URL FOR AGENT LICENSE #####
#####

Use this section to change the default Agent License URL.

<AgentLicense>
  <property name = "URL" value="Workload Automation SaaS agent license document"></property>
</AgentLicense>

</settings>

#####
##### SETTINGS FOR ALL Administrators users #####
#####

<settings role="Administrator">
  Put here setting to be applied only to users with Administrator role
</settings>

#####
##### SETTINGS FOR ALL Operators users #####
#####

<settings role="Operator">
</settings>

#####
##### SETTINGS FOR ALL Configurator users #####
#####

<settings role="Configurator">
</settings>

#####
##### SETTINGS FOR ALL Developer users #####
#####

<settings role="Developer">
</settings>

#####
##### SETTINGS FOR ALL Analyst users #####
#####

<settings role="Analyst">
</settings>

</tdwc>
```

For more information about how to customize global settings, see [Customizing your global settings on page 145](#).

Disable the What-if Analysis

You can disable the What-if Analysis in your environment by setting the **optman** `enWhatIf | wi` global option to *no* (default value is *yes*).

The `enWhatIf | wi` global option interacts with the `enWorkloadServiceAssurance | wa` global option, which enables or disables privileged processing of mission-critical jobs and their predecessors. For details about this interaction, see the following table.

Table 27. Interaction between `enWorkloadServiceAssurance` and `enWhatIf` global options

Options	Interaction
<code>enWorkloadServiceAssurance wa</code> is set to <i>yes</i> <code>enWhatIf wi</code> is set to <i>yes</i>	Both the Workload service assurance and the What-if Analysis features are fully enabled in your environment.
<code>enWorkloadServiceAssurance wa</code> is set to <i>yes</i> <code>enWhatIf wi</code> is set to <i>no</i>	The Workload service assurance is enabled. The What-if Analysis feature is disabled and an exception is issued if you try to use it.
<code>enWorkloadServiceAssurance wa</code> is set to <i>no</i> <code>enWhatIf wi</code> is set to <i>yes</i>	The Workload service assurance is partially enabled, just to allow the What-if Analysis feature to work properly. This means that: <ul style="list-style-type: none"> • The Workload service assurance is disabled and an exception is issued if you try to use it. • No critical job is added to the plan.
<code>enWorkloadServiceAssurance wa</code> is set to <i>no</i> <code>enWhatIf wi</code> is set to <i>no</i>	Both the Workload service assurance and the What-if Analysis features are disabled in your environment.

Configuring High Availability

How to configure, change, and share your settings repository.

Performance can be highly improved by configuring multiple Dynamic Workload Console instances in a High Availability configuration, so as to have multiple console instances working at the same time and with the same repository settings.

The Dynamic Workload Console is set to be always in High Availability and a front-end Network Dispatcher must be set up to handle and distribute all incoming session requests.

If you use a Dynamic Workload Console in High Availability configuration, when you connect to a Dynamic Workload Console you are not actually connecting to a specific console but to a load balancer that dispatches and redirects the connections among the nodes in the configuration. Therefore, for example, if a node fails, new user sessions are directed to other active nodes in the configuration and this change is completely transparent to users.

To perform this task you need to have access to an installed DB2®, Oracle, Informix® or MSSQL database. The High Availability cannot be configured using Apache Derby®

To implement this kind of configuration, the Administrator must ensure that the `datasource.xml`, located in the following path `opt/wa/DWC/DWC_DATA/usr/servers/dwcServer/configDropins/overrides`, has the same configuration on every Dynamic Workload Console in the cluster.

Configuring Dynamic Workload Console to view reports

This topic describes the configuration steps that you perform to be able to see the reports from the Dynamic Workload Console, if are using an Oracle database.

To access the databases where reports are stored, you must have the following prerequisites:

- A user ID and password to access the database
- A working connection between the Dynamic Workload Console and the database

Perform the following step on the system where the IBM Workload Scheduler engine is running:

- [Configuring for an Oracle database on page 166](#)

Configuring for an Oracle database

About this task

Actions taken on IBM Workload Scheduler engine:

For Oracle, the IT administrator, or the IBM Workload Scheduler IT administrator, or both working together, do the following:

1. Use the **TWS Oracle user** specified during the master domain manager installation or perform the following steps to create a new user:
 - a. Create a database user authorized to access the database and specify a password.
 - b. Launch the following script:

```
<TWA_home>/TWS/dbtools/Oracle/scripts/dbgrant.bat / sh
<ID_of_user_to_be_granted>
<database_name>
<database_admin_user> <password>
```

where the variables are as follows:

<TWA_home>

The IBM Workload Automation instance directory

<ID_of_user_to_be_granted>

The ID of the user created in step 1.a on page 166, who is going to be granted the access to the reports

<database_name>

The name of the database, as created when the master domain manager was installed

<database_schema_owner> <password>

The user ID and password of the database schema owner.

2. Define a valid connection string to the database:

a. Browse to the following path:

On Windows operating systems

```
<TWA_home>\usr\servers\engineServer\resources\properties
```

On UNIX operating systems

```
<TWA_DATA_DIR>/usr/servers/engineServer/resources/properties
```

b. Ensure that the following property is set in the `TWSConfig.properties` file to point to the Oracle JDBC URL:

```
com.ibm.tws.webui.oracleJdbcURL
```

For example:

```
com.ibm.tws.webui.oracleJdbcURL=
    jdbc:oracle:thin:@//9.132.235.7:1521/orcl
```

The Oracle JDBC URL is also to be specified in the **PARAM_DataSourceUrl** property in the `.\config\common.properties` file. The `common.properties` file is required when setting up for command line reporting. For more information about this file, see [Setting up for command line audit reporting on page 384](#) and the section about setting up for command line batch reporting in *IBM Workload Scheduler: User's Guide and Reference*.

c. Restart WebSphere Application Server Liberty Base .

Actions taken on the Dynamic Workload Console:

1. Log on to the Dynamic Workload Console.
2. In the navigation bar, select **Administration > Manage Engines**. The Manage Engines panels opens.
3. Select the engine you defined or create another engine. The Engine Connection properties panel is displayed.
4. In Database Configuration for Reporting, do the following:
 - a. Check **Enable Reporting** to enable the engine connection you selected to run reports.
 - b. In **Database User ID and Password**, specify the database user and password that you authorized to access reports.

Chapter 3. Configuring user authorization (Security file)

This chapter describes how to manage the authorizations to access scheduling objects assigned to IBM Workload Scheduler users.

Getting started with security

The way IBM Workload Scheduler manages security is controlled by a configuration file named **security file**. This file controls activities such as:

- Linking workstations.
- Accessing command-line interface programs and the Dynamic Workload Console.
- Performing operations on scheduling objects in the database or in the plan.

The security file for a fresh installation is located in the following path:

For a fresh installation of version 9.5.x or later

```
TWA_DATA_DIR
```

```
TWA_home\TWS
```

Upgraded environment originating from a version earlier than 9.5:

```
TWA_home/TWS
```

```
TWA_home\TWS
```

The security file contains some predefined access definitions:

- A full access definition for the user who installed the product, `<TWS_user>`.
- An access definition for the system administrator (root on UNIX™ or Administrator on Windows™).
- The following access definitions for the Dynamic Workload Console:
 - Analyst
 - Administrator
 - Operator
 - Developer

As you continue to work with the product, you might want to add more users with different roles and authorization to perform specific operations on a defined set of objects.

By default, the security model enabled when you perform a fresh installation is role-based. You can update your *security file* according to the role-based security model. The role-based security model allows you to update your *security file* with the security objects (domains, roles, and access control lists) that you define in the master domain manager database. You can define your security objects by using the **Manage Workload Security** interface from Dynamic Workload Console or the **composer** command-line program. The role-based security model is enabled through the setting of the **optman** `enRoleBasedSecurityFileCreation` global option. By default this option is set to `no`. To use the role-based security model, change the value to `yes`. For details about updating the security file according to the role-based security model, see [Role-](#)

[based security model on page](#) . For more information about the `enRoleBasedSecurityFileCreation` global option, see [Global options - detailed description on page 29](#).

If you are upgrading IBM Workload Scheduler version 9.3 or earlier, you might want to continue to use the classic security model that allows you to update the security file by using `dumpsec` and `makesec` commands from the command line. To continue to use the classic security model, the `enRoleBasedSecurityFileCreation` global option must be set to `no`. A new security file is then created and updated with the security objects (domains, roles, and access control lists) that you define in the master domain manager database by using the **Manage Workload Security** interface from Dynamic Workload Console or the **composer** command-line program. For details about updating the security file according to the classic security model, see [Classic security model on page](#)

Changes to `enRoleBasedSecurityFileCreation` global option are effective immediately. For details about the `enRoleBasedSecurityFileCreation` global option, see [Global options - detailed description on page](#) .



Note: The role-based security model and the classic security model are mutually exclusive.

Starting from version 9.5, Fix Pack 3, the term **\$SLAVES**, which applies to all fault-tolerant agents in both the classic and role-based security models, was replaced with the term **\$AGENTS** with the same scope. No change is required to your existing scripts nor environments.

Role-based security model

The security objects that you define by using the **Manage Workload Security** interface from Dynamic Workload Console, or the **composer** command-line program, are:

Access control lists

Each access control list is defined assigning roles to users or groups, on a specific security domain or folder.

Folders

Each folder has its own level of authorization that defines the set of actions that users or groups can perform on each folder.

Security roles

Each role represents a certain level of authorization and includes the set of actions that users or groups can perform.

Security domains

Each domain represents the set of scheduling objects that users or groups can manage.

You save the definitions of your security objects in the master domain manager database. If the role-based security model is enabled for your system (see [Getting started with security on page](#)), whenever you need to update the security objects, your *security file* is updated and converted into an encrypted format (for performance and security), replacing the previous file. The system uses this encrypted *security file* from that point onwards.

Each time a user runs IBM Workload Scheduler programs, commands, and user interfaces, the product compares the name of the user with the user definitions in the *security file* to determine if the user has permission to perform those activities, on the specified scheduling objects, in a certain security domain.

When the security file is updated on the master domain manager, the security settings on the master domain manager are automatically synchronized with the backup master domain manager.



Note: The role-based security model does not support centralized security management on fault-tolerant agents. On fault-tolerant agents, the security is managed locally on each workstation.

Configuring role-based security from Dynamic Workload Console

About this task

This section explains how to create and modify the security objects by using the **Manage Workload Security** interface from Dynamic Workload Console.

To create or modify security objects, you must have permission for the **modify** action on the object type **file** with attribute **name=security**.

When working with the role-based security from the Dynamic Workload Console, be aware that access to security objects is controlled by an "optimistic locking" policy. When a security object is accessed by user "A", it is not actually locked. The security object is locked only when the object update is saved by user "A", and then it is unlocked immediately afterward. If in the meantime, the object is accessed also by user "B", he receives a warning message saying that the object has just been updated by user "A", and asking him if he wants to override the changes made by user "A", or refresh the object and make his changes to the updated object.

Managing access control list

About this task

Create an access control list by assigning security roles to users or groups, in a certain security domain or in one or more folders.

You can:

- Give access to user or group.
- View access for user or group.
- View access for Security Domain or folders.
- Manage accesses.

Give access to user or group

About this task

To give access to users or groups complete the following procedure:

1. From the navigation toolbar, click **Administration**.
2. In the **Workload Environment Design**, select **Manage Workload Security**.

Result

The Manage Workload Security panel opens.

3. From the drop-down list, select the IBM Workload Scheduler engine on which you want to manage security settings.
4. In the Access Control List section, click **Give access to user or group**.

Result

The Create Access Control List panel opens.

5. Enter the user name or the group name, the assigned roles, and the security domain or enter the folders assigned. For each Access Control List you can associate one or more folders.
6. Click **Save** to save the access definition in the database.
7. Click **Save and Create New** to save the access definition in the database and proceed to create a new access definition.
8. Click **Save and Exit** to save the access definition in the database and return to the Manage Workload Security panel.

Results

The access definition has now been added to the database. If the **optman** `enRoleBasedSecurityFileCreation` global option is set to `yes`, the access definition is activated in your security file.

View access for user or group

About this task

From Manage Workload Security, you can also view the access for users or groups.

1. In the Access Control List section of the Manage Workload Security panel, click **View access for user or group**.

Result

The input field for the user or group name is displayed.

2. Enter the user or group name and click **View**.

Result

The user or group access, with the assigned roles, to the related security domains is displayed.

View access for Security Domain

About this task

From Manage Workload Security, you can also view the access to a certain security domain.

1. In the Access Control section of the Manage Workload Security panel, click **View access for Security Domain**.

Result

The input field for the security domain name is displayed.

2. Enter the security domain name and click **View**.

Result

The list of users or groups, with the assigned roles, that have access to the specified security domain is displayed.

Manage accesses

About this task

From Manage Workload Security, you can also **remove** and **edit** existing access control lists.

1. In the Access Control List section of the Manage Workload Security panel, click **Manage Accesses**.

Result

The list of users or groups, with the assigned roles, that have access to the different security domains is displayed.

2. Select the access control list that you want to manage.
3. Select the action that you want to run on the selected access control list.

If you select the **edit** action, you can change only the roles associated with the access control list. You cannot change the associated domain. If you want to change the domain, you must **remove** the access control list and redefine the access control list with a new domain.

Managing folders

About this task

Folders help you to organize jobs and job streams into different categories. You can create folders with different levels of authorization that define the set of actions that users or groups can perform on each folder. More than one folder can be associated to the same Access Control List, and the level of security is also applied to the sub-folders.

You can also grant a user administrator privileges on a folder and its sub-folders so that this user can then create access control lists, with a dedicated role to manage the objects contained in the folder. See [Granting administrator permissions to a user on a folder on page 173](#).

Creating, renaming, or deleting a folder

About this task

To create, rename, or delete a folder:

1. From the navigation toolbar, click **Administration**.
2. In the **Security**, select **Manage Workload Security**.
3. From the drop-down list, select the IBM Workload Scheduler engine on which you want to manage security settings.

Result

The Manage Workload Security panel opens.

4. In the folders section, click **Manage Folder**.

Result

The **Manage Folders** panel opens. From this panel you can:

- Use the search box to search folders and job streams in the current view.
- Create a folder or subfolder, rename or delete a folder.

Granting administrator permissions to a user on a folder

About this task

The IBM Workload Scheduler administrator can grant administrator permissions to a user on a folder so that the user can freely define access control lists for other users on the same folder or any sub-folders. Users can then access the objects in the folder or sub-folders in accordance with the access permissions they have on the objects.

 **Tip:** Users with the FULLCONTROL security role assigned automatically have administrator rights on folders.

The following scenario demonstrates how Tim, the IBM Workload Scheduler administrator, grants Linda, the application administrator (`appl_admin` user), permissions on the folder named `/PRD/APP1/`, and how Linda grants access to Alex, the application user, to work with the objects defined in `/PRD/APP1/FINANCE`:

1. Tim, the IBM Workload Scheduler administrator, grants Linda, the `appl_admin` user, administrator permissions on the folder, `/PRD/APP1/`, through the definition of an access control list and by modifying her currently assigned role, `APPADMIN`. Optionally, Tim can create a new role with the appropriate permissions to achieve the same result.
 - a. From the **Manage Workload Security** page, Tim selects **Manage roles**.
 - b. He then selects her current role from the list, `APPADMIN` and clicks **Edit**.
 - c. He gives this role administrator permissions on folders by selecting **Delegate folder permission (folder - acl)** in the **Administrative Tasks** section and clicks **Save and Exit**.
 - d. Tim then creates an access control list for Linda, the `appl_admin` user. From the **Manage Workload Security** page, Tim selects **Give access to users or groups**.
 - e. From the **Create Access Control List** page, Tim selects **User name** from the drop-down and enters Linda's user name, `appl_admin` in the text box.
 - f. In the **Role** text box, Tim enters the `APPADMIN` role he modified earlier.
 - g. In the text box next to the **Folder** selection, Tim enters the folder path of the folders on which he wants to grant Linda permissions, `/PRD/APP1/`.

Result

Linda, the `appl_admin` user, with the `APPADMIN` role assigned, can now access the entire `/PRD/APP1/` hierarchy, can create new folders in this path, and can assign access to these folders to other users.

2. Linda needs to give application users such as Alex, access to the objects in the `/PRD/APP1/FINANCE` folder. She creates a new access control list on the folder for the application user and assigns a role to this user.
 - a. From the **Manage Workload Security** page, Linda selects **Give access to users or groups** from the **Access Control List** section.
 - b. On the **Create Access Control List** page, Linda selects **User name** from the drop-down and enters the user name for Alex, the application user, `appl_user`.
 - c. Since Linda cannot create new roles, she specifies an existing role in the Role text box. Only Tim, the IBM Workload Scheduler administrator, can create new roles.
 - d. In the text box next to the **Folder** selection, Linda enters the folder path to the new sub-folder she created and to which Alex requires access: `/PRD/APP1/FINANCE`.

Result

Alex now can access the `/PRD/APP1/FINANCE` folder. He does not have access permissions on the `/PRD/APP1` folder.

Managing security domains

About this task

A security domain represents the set of objects that users or groups can manage. For example, you can define a domain that contains all objects named with a prefix 'AA'. If you want to specify different security attributes for some or all of your users, you can create additional security domains based on specific matching criteria.

You can filter objects by specifying one or more attributes for each security object type. You can include or exclude each attribute from the selection. For example, you can restrict access to a set of objects having the same name or being defined on the same workstation, or both.

For the attributes that you can specify for each security object type, see [Attributes for object types on page](#) .

For the values that you can specify for each object attribute, see [Specifying object attribute values on page](#) .

You can create new security domains or manage existing security domains.

Create new security domain

About this task

To create a new security domain from the Dynamic Workload Console, complete the following procedure:

1. From the navigation toolbar, click **Administration**.
2. In the **Security**, select **Manage Workload Security** .

Result

The Manage Workload Security panel opens.

3. From the drop-down list, select the IBM Workload Scheduler engine on which you want to manage security settings.
4. In the Security Domains section, click **Create new Security Domain**.

Result

The security domain creation panel opens.

5. Enter the name of the security domain that you are creating and, optionally, the domain description.
6. Select the type of security domain that you want to define:

Simple

To define a filtering rule that applies to all object types. Events and actions are excluded from this filtering rule.

Complex

To define different filtering rules for different object types.

7. Use object filtering to select the set of security objects that users or groups can manage in the security domains that you are defining. You can use the wildcard character (*) when defining object attributes.
8. Click **View** to see the mapping between the set of security objects that you are assigning to the domain and the corresponding set of security objects in the classic security model.

9. Click **Save** to save the security domain definition in the database.
10. Click **Save and Exit** to save the security domain definition in the database and then exit.

Results

The security domain has now been added to the database. If the `optmanenRoleBasedSecurityFileCreation` global option is set to `yes`, the security domain is activated in your security file.

Manage security domain

About this task

From Manage Workload Security, you can also **remove**, **edit**, and **duplicate** existing security domains.

1. In the Security Domains section of the Manage Workload Security panel, click **Manage Security Domain**.

Result

The list of the available security domains is displayed.

2. Select the security domains that you want to manage.
3. Select the action that you want to run on the selected security domains.

Managing security roles

About this task

A security role represents a certain level of authorization and includes the set of actions that users or groups can perform on a set of object types.

For the list of actions that users or groups can perform on the different objects, for each IBM Workload Scheduler task, see [Actions on security objects on page 185](#).

A set of predefined security roles is available in the master domain manager database after the product has been installed:

- A full access definition for the user who installed the product, TWS_user with the default security role assigned named `FULLCONTROL`.
- An access definition for the system administrator, root on UNIX or Administrator on Windows.

You can create new security roles or manage existing security roles.

Create new role

About this task

To create a new security role from the Dynamic Workload Console, complete the following procedure:

1. From the navigation toolbar, click **Administration**.
2. In the **Security** select **Manage Workload Security**.

Result

The Manage Workload Security panel opens.

3. From the drop-down list, select the IBM Workload Scheduler engine on which you want to manage security settings.
4. In the Roles section, click **Create new role**.

Result

The Create Role panel opens.

5. Enter the name of the security role that you are creating and, optionally, the role description.
6. For each of the IBM Workload Scheduler tasks, assign the level of access for performing certain actions on specific object types to the security role. You can assign a predefined or a custom level of access.
7. Click **Show Details** to see the permissions associated to a predefined level of access, or to define your custom level of access. Tooltips are available to explain what a certain permission means for a particular object type.
8. Click **View** to see the mapping between the set of permissions that you are assigning and the corresponding set of permissions in the classic security model.
9. Click **Save** to save the security role definition in the database.
10. Click **Save and Exit** to save the security role definition in the database and return to the Manage Workload Security panel.

Results

The security role has now been added to the database. If the **optman** `enRoleBasedSecurityFileCreation` global option is set to `yes`, the security role is activated in your security file.

Manage roles

About this task

From Manage Workload Security, you can also **remove**, **edit**, and **duplicate** existing roles.

1. In the Roles section of the Manage Workload Security panel, click **Manage roles**.

Result

The list of the available security roles is displayed.

2. Select the security roles that you want to manage.
3. Select the action that you want to run on the selected roles.

Configuring role-based security with composer command-line

About this task

This section explains how to create or modify the security objects in the database, by using the **composer** command line interface.

To define security objects in the database, see:

[Access control list definition on page 177](#)

[Security domain definition on page 178](#)

[Security role definition on page 181](#)

To manage security objects in the database, see the section about managing objects with composer command-line, in the *User's Guide and Reference*.

To define or modify security objects, you must have permission for the **modify** action on the object type **file** with attribute **name=security**.

Security access control list definition

In the role-based security model, an access control list assigns security roles to users or groups, in a certain security domain or on a specific folder or folder hierarchy. You can include multiple security access control list definitions in the same text file, along with security domain definitions and security role definitions.

Each security access control list definition has the following format and arguments:

Syntax

```
accesscontrollist for security_domain_name
    user_or_group_name [security_role, security_role]...
    [user_or_group_name [security_role, security_role]...]...
end
```

[**securitydomain** ...]

[**securityrole** ...]

```
accesscontrollist folder folder_name
    user_or_group_name [security_role, security_role]...
    [user_or_group_name [security_role, security_role]...]...
end
```

Arguments

security_domain_name

Specifies the name of the security domain on which you are defining the access control list.

***user_or_group_name* [*security_role*, *security_role*]**

Assigns one or more security roles to a certain user or group, on the specified security domain.

folder_name

Specifies the name of the folder to which you can associate an access control list. If the access control list is associated to a folder, then the security roles are valid for all of the objects contained in the folder. When specifying folder names, ensure you include a forward slash (/) before the folder name. Include a forward slash after the folder name to indicate that the access control list is defined only on the folder specified, excluding any sub-folders. A folder name without a final forward slash indicates that the access control list is defined on the folder, as well as on any sub-folders.

Associating an access control list to a folder is a quick and easy method to grant access to all of the objects defined in a folder. If, instead, you need to restrict access to a subset of objects in the folder (for example, objects with a certain name, or specific userlogon, cpu or jcl), then using an access control list associated to a security domain is more effective. With security domains you can filter objects by specifying one or more attributes for each security object type.

See the following composer commands documented in the *User's Guide and Reference* when working with folders: Chfolder, Listfolder, Mkfolder, Rmfolder, and Renamefolder.

Example

Examples

The following example defines:

- An access control list on the `SECDOM1` domain
- An access control list on `SECDOM2` domain
- An access control list on the folder `/FOL1/FOL2/`
- An access control list on the folder `/APPS/APP1` and any sub-folders, if present, for example, `/APPS/APP1/APP1A`.

```
ACCESSCONTROLLIST FOR SECDOM1
  USER1 SECR0LE1, SECR0LE2, SECR0LE3
  USER2 SECR0LE4
  USER3 SECR0LE2, SECR0LE4
END

ACCESSCONTROLLIST FOR SECDOM2
  USER1 SECR0LE1, SECR0LE2
  USER2 SECR0LE3
END

ACCESSCONTROLLIST FOLDER /FOL1/FOL2/
  USER1 SECR0LE1
END

ACCESSCONTROLLIST FOLDER /APPS/APP1
  USER1 SECR0LE1
END
```

Security domain definition

In the role-based security model, a security domain represents the set of objects that users or groups can manage. For example, you can define a domain that contains all objects named with a prefix 'AA'. If you want to specify different security attributes for some or all of your users, you can create additional security domains based on specific matching criteria. You can filter objects by specifying one or more attributes for each security object type. You can include or exclude each attribute from the selection. For example, you can restrict access to a set of objects having the same name or being defined on the same workstation, or both.

You can include multiple security domain definitions in the same text file, along with security role definitions and access control list definitions.

By default, a security domain named ALLOBJECTS is available. It contains all scheduling objects and cannot be renamed nor modified.

Each security domain definition has the following format and arguments:

Syntax

Each attribute can be included or excluded from the selection using the plus (+) and tilde (~) symbols.

```
securitydomain security_domain_name
  [description "description"]
  [common [[+|~]object_attribute [= value | @[, value | @]...]]]
  [object_type [[+|~]object_attribute [= value | @[, value | @]...]]]
  [object_type [[+|~]object_attribute [= value | @[, value | @]...]]]...
end
[securityrole ...]
[accesscontrollist ...]
```

Arguments

securitydomain *security_domain_name*

Specifies the name of the security domain. The name must start with a letter, and can contain alphanumeric characters, dashes, and underscores. It can contain up to 16 characters.

description "*description*"

Provides a description of the security domain. The description can contain up to 120 alphanumeric characters. The text must be enclosed within double quotes.

common [[+|~]*object_attribute* [= *value* | @[, *value* | @]...]]

Provides object attributes that are common to all the security object types.

object_type [[+|~]*object_attribute* [= *value* | @[, *value* | @]...]]

For each object type, specifies the attributes that apply to that object type and the related values. Each attribute can be included or excluded from the selection using the plus (+) and tilde (~) symbols. Wildcard (@) is supported for the attribute value: *object_attribute* = @ means that all the objects matching the object attribute must be included in the domain. For the use of wildcard (@), see the examples below.

For the attributes that you can specify for each security object type, see the section about managing security with the Dynamic Workload Console, in the *Dynamic Workload Console User's Guide*.

For the values that you can specify for each object attribute, see the section about managing security with the Dynamic Workload Console, in the *Dynamic Workload Console User's Guide*.

Example

Examples

The following example defines a security domain named `SECDOM1` and a security domain named `SECDOM2`:

```

securitydomain SECDOM1
description "Sample Security Domain1"
job      cpu =  $THISCPU, # The workstation where the user logs on
          $MASTER, # The master workstation
          $AGENTS, # Any fault tolerant agent
          $REMOTES # Any standard agent
          cogs@   # Any workstation whose name starts with "cogs"
+ folder = / # Jobs defined in any folder
+ cpufolder = / # Workstations defined in any folder
+ name = A@   # Any job whose name starts with "A"
~ name = A2@  # but doesn't start with A2
+ jcltype = SCRIPTNAME # Allow only SCRIPTNAME type of job definition
+ jcltype = DOCCOMMAND # Allow only DOCCOMMAND type of job definition
+ logon = $USER, # Streamlogon is the conman/composer user
          $OWNER, # Streamlogon is the job creator
          $JCLOWNER, # Streamlogon is the OS owner of the file
          $JCLGROUP # Streamlogon is the OS group of the file
~ logon = root, twsuser # The job cannot logon as "root" or "twsuser"
+ jcl = "/usr/local/bin/@" # The jobs whose executable file that is
present in /usr/local/bin
~ jcl = "@rm@" # but whose JSDL definition does not contain the
string "rm"
end

securitydomain SECDOM2
description "Sample Security Domain2"
common      cpu=@+name=@
userobj     cpu=@ + cpufolder = /
job         cpu=@+ folder = / + cpufolder = /
schedule   cpu=@+name=AP@+ folder = / + cpufolder = /
resource    cpu=@ + folder = / + + cpufolder = /
prompt      folder = /
file        name=@
cpu         cpu=@ + folder = /
parameter  cpu=@ + folder = / + cpufolder = /
calendar    folder = /
report      name=@
eventrule   name=@ + folder = /
action      provider=@
event       provider=@
varitable   name=@ + folder = /
wkldapp     name=@ + folder = /
runcygrp    name=@ + folder = /
lob         name=@
folder      name=/
end

```

Security role definition

In the role-based security model, a security role represents a certain level of authorization and includes the set of actions that users or groups can perform. You can include multiple security role definitions in the same text file, along with security domain definitions and access control list definitions.

Each security role definition has the following format and arguments:

Syntax

```
securityrole security_role_name
  [description "description"]
  object_type access[=action[,action]...]
  [object_type access[=action[,action]...]...]
end

[securitydomain ...]

[accesscontrollist ...]
```

Arguments

securityrole*securityrolename*

Specifies the name of the security role. The name must start with a letter, and can contain alphanumeric characters, dashes, and underscores. It can contain up to 16 characters.

description "description"

Provides a description of the security role. The description can contain up to 120 alphanumeric characters. The text must be enclosed within double quotes.

object_type access[=action[,action]...]

For each object type, specifies a list of actions that users or groups can perform on that specific object type.

[Table 28: Security object types on page 181](#) shows the different object types and how they are referenced with **composer** and with the Dynamic Workload Console:

Table 28. Security object types

Object type - composer	Object type - Dynamic Workload Console	Description
action	Actions	Actions defined in scheduling event rules
calendar	Calendars	User calendars
cpu	Workstations	Workstations, domains, and workstation classes
event	Events	Event conditions in scheduling event rules

Table 28. Security object types
(continued)

Object type - composer	Object type - Dynamic Workload Console	Description
eventrule	Event Rules	Scheduling event rule definitions
file	Files	IBM Workload Scheduler database files
folder	Folders	The folder within which jobs and job streams are defined.
job	Jobs	Scheduled jobs and job definitions
lob	IBM Application Lab	IBM Application Lab
parameter	Parameters	Local parameters
prompt	Prompts	Global prompts
report	Reports	The following reports in Dynamic Workload Console: RUNHIST Job Run History RUNSTATS Job Run Statistics WWS Workstation Workload Summary WWR Workstation Workload Runtimes SQL Custom SQL ACTPROD Actual production details (for current and archived plans) PLAPROD Planned production details (for trial and forecast plans)
resource	Resources	Scheduling resources
runcygrp	Run Cycle Groups	Run cycle groups
schedule	Job Streams	Job streams
userobj	User Objects	User objects

Table 28. Security object types
(continued)

Object type - composer	Object type - Dynamic Workload Console	Description
variable	Variable Tables	Variable tables
wkldappl	Workload Application	Workload application

Table 29: Actions that users or groups can perform on the different objects on page 183 shows the actions that users or groups can perform on the different objects.

Table 29. Actions that users or groups can perform on the different objects

Actions that users or groups can perform on the different objects				
acl	deldep	modify	stop	
add	delete	release	submit	
adddep	display	reply	submitdb	
altpass	fence	rerun	unlink	
altpri	kill	resetfta	unlock	
build	limit	resource	use	
cancel	link	run		
confirm	list	shutdown		
console	manage	start		

For the actions that users or groups can perform on a specific object type, for each of the IBM Workload Scheduler task, see the section about managing security roles with the Dynamic Workload Console, in the *Dynamic Workload Console User's Guide*.

Example

Examples

The following example defines security role `SECROLE1` and security role `SECROLE2`:

```
SECURITYROLE SECROLE1
DESCRIPTION "Sample Security Role"
SCHEDULE ACCESS=ADD,ADDDEP,ALTPRI,CANCEL,DELDEP,DELETE,
          DISPLAY,LIMIT,MODIFY,
          RELEASE
```

```

RESOURCE      ACCESS=ADD,DELETE,DISPLAY,MODIFY,RESOURCE,USE,LIST,UNLOCK
PROMPT        ACCESS=ADD,DELETE,DISPLAY,MODIFY,REPLY,USE,LIST,UNLOCK
FILE          ACCESS=BUILD,DELETE,DISPLAY,MODIFY,UNLOCK
FOLDER        ACCESS=ADD,DELETE,DISPLAY,MODIFY,USE,LIST,UNLOCK,ACL
CPU           ACCESS=LIMIT,LINK,MODIFY,SHUTDOWN,START,STOP,UNLINK,LIST,UNLOCK,RUN
PARAMETER     ACCESS=ADD,DELETE,DISPLAY,MODIFY,LIST,UNLOCK
CALENDAR      ACCESS=ADD,DELETE,DISPLAY,MODIFY,USE,LIST,UNLOCK
REPORT        ACCESS=DISPLAY
EVENTRULE     ACCESS=ADD,DELETE,DISPLAY,MODIFY,LIST,UNLOCK
ACTION        ACCESS=DISPLAY,SUBMIT,USE,LIST
EVENT         ACCESS=USE
VARIABLE      ACCESS=ADD,DELETE,DISPLAY,MODIFY,USE,LIST,UNLOCK
WKLDAPPL      ACCESS=ADD,DELETE,DISPLAY,MODIFY,LIST,UNLOCK
RUNCYGRP      ACCESS=ADD,DELETE,DISPLAY,MODIFY,USE,LIST,UNLOCK
LOB           ACCESS=USE
END

SECURITYROLE SECR0LE2
DESCRIPTION "Sample Security Role"
SCHEDULE      ACCESS=ADD,ADDDEP,ALTPRI,CANCEL,DELDEP,DELETE,
              DISPLAY,LIMIT,MODIFY,
RELEASE
RESOURCE      ACCESS=ADD,DELETE,DISPLAY,MODIFY,RESOURCE,USE,LIST,UNLOCK
PROMPT        ACCESS=ADD,DELETE,DISPLAY,MODIFY,REPLY,USE,LIST,UNLOCK
END

```

The following example defines a new security role `APP_ADMIN`, for the user `APP1_ADMIN` and assigns administrator permissions on the folder hierarchy `/PRD/APP1/`, so that the `APP1_ADMIN` user can create access control lists to give other users access to the objects in this folder or its sub-folders:

Security role definition

```

SECURITYROLE APP_ADMIN
DESCRIPTION "Security Role"
JOB    ADD,MODIFY,SUBMITDB,USE,ADDDEP,RUN,RELEASE,REPLY,DELETE,DISPLAY,
      CANCEL,SUBMIT,CONFIRM,RERUN,LIST,DELDEP,KILL,UNLOCK,ALTPRI
SCHEDULE  ADD,ADDDEP,ALTPRI,CANCEL,DELDEP,DELETE,DISPLAY,LIMIT,MODIFY,RELEASE
FOLDER    ADD,DELETE,DISPLAY,MODIFY,USE,LIST,UNLOCK,ACL

```

Security file

```

USER APP_ADMINofAPP1
CPU=@+LOGON="APP_ADMIN"
BEGIN
JOB    FOLDER="/PRD/APP1/", "/PRD/APP1" + CPUFOLDER = / ACCESS=ADD,ADDDEP,
      ALTPRI,CANCEL,SUBMIT,
      CONFIRM,RERUN,LIST,DELDEP,KILL,UNLOCK,ALTPRI
SCHEDULE  FOLDER="/PRD/APP1/", "/PRD/APP1" + CPUFOLDER = / ACCESS=ADD,ADDDEP,
      ALTPRI,CANCEL,DELDEP,
      DELETE,DISPLAY,LIMIT,MODIFY,RELEASE
FOLDER  NAME="/PRD/APP1/", "PRD/APP1" ACCESS=ADD,DELETE,DISPLAY,MODIFY,USE,
      LIST,UNLOCK,ACL

```


Actions on security objects

The following tables show the actions that users or groups can perform on the different object types, for each IBM Workload Scheduler task. See in parenthesis the corresponding *actions* and *objects* values that you must use when defining role-based security with **composer** command line interface.

Table 30. Actions that users or groups can perform when designing and monitoring the workload

Design and Monitor Workload	
Actions that users or groups can perform	Security object types
List (list)	Jobs (job)
Display (display)	Job Streams (schedule)
Create (add)	User Objects (userobj)
Delete (delete)	Prompts (prompt)
Modify (modify)	Resources (resource)
Use (use)	Calendars (calendar)
Unlock (unlock)	Run Cycle Groups (runcygrp)
Actions on remote workstations while modeling jobs (cpu-run)	Variable Tables (vartable)
	Workload Application (wkldappl)
	Workflow Folders (folder)
	Parameters (parameter)


 **Note:** See in parenthesis the corresponding *actions* and *objects* values that you must use when defining role-based security with the **composer** command-line interface.

Table 31. Actions that users or groups can perform when modifying current plan

Modify current plan
Actions that users or groups can perform on the current plan
Add job stream dependency (schedule - adddep)
Add job dependency (job - adddep)
Remove job dependency (job - deldep)
Remove job stream dependency (schedule - deldep)
Change job priority (job - altpri)

Table 31. Actions that users or groups can perform when modifying current plan
(continued)

Modify current plan
Actions that users or groups can perform on the current plan
Change job stream priority (schedule - altpri)
Cancel job (job - cancel)
Cancel job stream (schedule - cancel)
Rerun job (job - rerun)
Confirm job (job - confirm)
Release job (job - release)
Release job stream (schedule - release)
Kill jobs (job - kill)
Reply to prompts (prompt - reply)
Reply to job prompts (job - reply)
Reply to job stream prompts (schedule - reply)
Alter user password (userobj - altpass)
Change jobs limit (schedule - limit)
Actions on job remote system (job - run)
Change resource quantity (resource - resource)



Note: See in parenthesis the corresponding *actions* and *objects* values that you must use when defining role-based security with the **composer** command line interface.

Table 32. Actions that users or groups can perform when submitting workload

Submit Workload
Workload definitions that can be added to the current plan
Only existing job definitions (job - submitdb)
Existing jobs definitions and ad hoc jobs (job - submit)
Existing job stream definitions (schedule - submit)

Table 32. Actions that users or groups can perform when submitting workload
(continued)

Submit Workload

Workload definitions that can be added to the current plan



Note: See in parenthesis the corresponding *actions* and *objects* values that you must use when defining role-based security with the **composer** command line interface.

Table 33. Actions that users or groups can perform when managing the workload environment

Manage Workload Environment

Actions that users or groups can perform on workstations, domains, and workstation classes

List workstations (cpu - list)

Display workstation details (cpu - display)

Create workstations (cpu - add)

Delete workstations (cpu - delete)

Modify workstations (cpu - modify)

Use workstations (cpu - use)

Unlock workstations (cpu - unlock)

Start a workstation (cpu - start)

Stop a workstation (cpu - stop)

Change limit (cpu - limit)

Change fence (cpu - fence)

Shutdown (cpu - shutdown)

Reset FTA (cpu - resetfta)

Link (cpu - link)

Unlink (cpu - unlink)

Use 'console' command from conman (cpu - console)

Upgrade workstation (cpu - manage)

Table 33. Actions that users or groups can perform when managing the workload environment
(continued)

Manage Workload Environment

Actions that users or groups can perform on workstations, domains, and workstation classes



Note: See in parenthesis the corresponding *actions* and *objects* values that you must use when defining role-based security with the **composer** command line interface.

Table 34. Actions that users or groups can perform when managing event rules

Manage Event Rules

Actions that users or groups can perform on event rules

List event rules (eventrule - list)

Display event rules details (eventrule - display)

Create event rules (eventrule - add)

Delete event rules (eventrule - delete)

Modify event rules (eventrule - modify)

Use event rules (eventrule - use)

Unlock event rules (eventrule - unlock)

Display actions in the event rules (action - display)

Monitor triggered actions (action - list)

Use action types in the event rules (action - use)

Submit action (action - submit)

Use events in the event rules (event - use)

Use a File Monitor event on the workstation where the file resides. (event - display)

Table 34. Actions that users or groups can perform when managing event rules
(continued)

Manage Event Rules

Actions that users or groups can perform on event rules



Note: See in parenthesis the corresponding *actions* and *objects* values that you must use when defining role-based security with the **composer** command line interface.

Table 35. Administrative tasks that users or groups can perform

Administrative Tasks

Administrative tasks that users or groups can perform

View configuration (dump security and global options) (file - display)

Change configuration (makesec, optman add) (file - modify)

Delete objects definitions (file - delete)

Unlock objects definitions (file - unlock)

Allow planman deploy, prodsked and stageman (file - build)

Delegate security on folders (folder - acl)



Note: See in parenthesis the corresponding *action* and *object* values that you must use when defining role-based security with the **composer** command-line interface.

Table 36. Actions that users or groups can perform on workload reports

Workload Reports

Actions that users or groups can perform on workload reports

Generate workload reports (display report)	Reports in Dynamic Workload Console RUNHIST Job Run History RUNSTATS Job Run Statistics WWS Workstation Workload Summary WWR Workstation Workload Runtimes
---	---

Table 36. Actions that users or groups can perform on workload reports
(continued)

Workload Reports	
Actions that users or groups can perform on workload reports	
SQL	Custom SQL
ACTPROD	Actual production details (for current and archived plans)
PLAPROD	Planned production details (for trial and forecast plans)



Note: See in parenthesis the corresponding *actions* and *objects* values that you must use when defining role-based security with the **composer** command line interface.

Table 37. Actions that users or groups can perform on Application Lab

Application Lab	
Actions that users or groups can perform on Application Lab	
Access Application Lab (use lob)	



Note: See in parenthesis the corresponding *actions* and *objects* values that you must use when defining role-based security with **composer** command line interface.

Table 38. Actions that users or groups can perform on folders.

Folders	
Actions that users or groups can perform on folders	
Access folders	
chfolder (display)	
listfolder (list or list and display)	
mkfolder (modify)	
rmfolder (delete)	

Table 38. Actions that users or groups can perform on folders.

(continued)

Folders
Actions that users or groups can perform on folders
renamefolder (add)



Note: See in parenthesis the corresponding *actions* and *objects* values that you must use when defining role-based security with the **composer** command line interface.

Attributes for security object types

[Table 39: Attributes for security object types on page 191](#) shows the attributes that you can specify for each security object type (see in parenthesis the corresponding object type and object attribute that you must use when defining security objects with the **composer** command line interface).

Table 39. Attributes for security object types

Security object type	Attribute Name (name)	Workstation (cpu)	Custom (custom)	JCL (jcl)	JCLtype (jcltype)	Logon (logon)	Provider (provider)	Type (type)	Host (host)	Port (port)	Folder (folder)	CPU Folder (cpu folder)
Actions (action)							✓	✓	✓	✓		
Calendars (calendar)	✓										✓	
Workstations (cpu)	✓							✓			✓	
Events (event)			✓				✓	✓				
Event rules (eventrule)	✓										✓	
Files (file)	✓											
Jobs (job)	✓	✓		✓	✓	✓					✓	✓
Application Lab (lob)	✓											
Parameters (parameter)	✓	✓									✓	✓
Prompts (prompt)	✓										✓	
Reports (report)	✓											
Resource (resource)	✓	✓									✓	
RunCycle groups (runcygrp)	✓										✓	
Job streams (schedule)	✓	✓									✓	✓
User objects (userobj)		✓				✓						✓

Table 39. Attributes for security object types (continued)

Security object type	Attribute Name (name)	Workstation (cpu)	Custom (custom)	JCL (jcl)	JCLtype (jcltype)	Logon (logon)	Provider (provider)	Type (type)	Host (host)	Port (port)	Folder (folder)	CPU Fol der (cpu folder)
Variable tables (variable)	✓										✓	
Workload applications (wkldappl)	✓										✓	
Folders (folder)	✓											

For the values that are allowed for each object attribute, see [Specifying object attribute values on page 192](#).

Specifying object attribute values

The following values are allowed for each object attribute (see in parenthesis the corresponding object type and object attribute for the **composer** command line interface):

Name (name)

Specifies one or more names for the object type.

- For the Files (file) object type, the following values apply:

globalopts

Allows the user to set global options with the `optman` command. The following access types are allowed:

- Display access for `optman ls` and `optman show`
- Modify access for `optman chg`

prodsked

Allows the user to create, extend, or reset the production plan.

security

Allows the user to manage the security file.

Symphony

Allows the user to run **stageman** and **JnextPlan**.

trialsked

Allows the user to create trial and forecast plans or to extend trial plans.



Note: Users who have restricted access to files should be given at least the following privilege to be able to display other object types that is, Calendars (calendar) and Workstations (cpu):



```
file name=globalopts action=display
```

- For the **Variable Tables (vartable)** object type, you can use the \$DEFAULT value for the **Name (name)** attribute to indicate the default variable table. This selects the table that is defined with the `isdefault` attribute.

Workstation (cpu)

Specifies one or more workstation, domain, or workstation class name. Workstations and workstation classes can optionally be defined in a folder. If this attribute is not specified, all defined workstations and domains can be accessed. Workstation variables can be used:

\$MASTER

The IBM Workload Scheduler master domain manager.

\$AGENTS

Any fault-tolerant agent.

\$REMOTES

Any standard agent.

\$THISCPU

The workstation on which the user is running the IBM Workload Scheduler command or program.

If you use the **composer** command line to define security domains, the following syntax applies:

```
cpu=[folder/]workstation[, [folder/]workstation]...
```

folder=foldername

Scheduling objects such as, jobs, job streams, and workstations, to name a few, can be defined in a folder. A folder can contain one or more scheduling objects, while each object can be associated to only one folder. The default folder is the root folder (/).

cpufolder=foldername

The folder within which the workstation or workstation class is defined.

Custom (custom)

Use this attribute to assign access rights to events defined in event plug-ins. The precise syntax of the value depends on the plug-in. For example:

- Specify different rights for different users based on SAP R/3 event names when defining event rules for SAP R/3 events.
- Define your own security attribute for your custom-made event providers.
- Specify the type of event that is to be monitored. Every event can refer to an event provider.

If you use **composer** command line to define security domains, the following syntax applies:

```
custom=value[,value]...
```

JCL (jcl)

Specifies the command or the path name of a job object's executable file. If omitted, all defined job files and commands qualify.

You can also specify a string that is contained in the task string of a JSDL definition to be used for pattern matching.

If you use **composer** command line to define security domains, the following syntax applies:

```
jcl="path" | "command" | "jSDL"
```

JCL Type (jcltype)

Specifies that the user is allowed to act on the definitions of jobs that run only scripts (if set to scriptname) or commands (if set to docommand). Use this optional attribute to restrict user authorization to actions on the definitions of jobs of one type only. Actions are granted for both scripts and commands when JCL Type (jcltype) is missing.

A user who is not granted authorization to work on job definitions that run either a command or a script is returned a security error message when attempting to run an action on them.

If you use **composer** command line to define security domains, the following syntax applies:

```
jcltype=[scriptname | docommand]
```

Logon (logon)

Specifies the user IDs. If omitted, all user IDs qualify.

You can use the following values for the **Logon (logon)** attribute to indicate default logon:

\$USER

Streamlogon is the conman/composer user.

\$OWNER

Streamlogon is the job creator.

\$JCLOWNER

Streamlogon is the OS owner of the file.

\$JCLGROUP

Streamlogon is the OS group of the file.

If you use **composer** command line to define security domains, the following syntax applies:

```
logon=username[,username]...
```

Provider (provider)

For **Actions (action)** object types, specifies the name of the action provider.

For **Events (event)** object types, specifies the name of the event provider.

If **Provider (provider)** is not specified, no defined objects can be accessed.

If you use **composer** command line to define security domains, the following syntax applies:

```
provider=provider_name[,provider_name]...
```

Type (type)

For **Actions (action)** object types, is the `actionType`.

For **Events (event)** object types, is the `eventType`.

For **Workstations (cpu)** object types, the permitted values are those used in composer or the Dynamic Workload Console when defining workstations, such as `manager`, `broker`, `fta`, `agent`, `s-agent`, `x-agent`, `rem-eng`, `pool`, `d-pool`, `cpuclass`, and `domain`.



Note: The value `master`, used in conman is mapped against the `manager` security attributes.

If **Type (type)** is not specified, all defined objects are accessed for the specified providers (this is always the case after installation or upgrade, as the type attribute is not supplied by default).

If you use **composer** command line to define security domains, the following syntax applies:

```
type=type[,type]...
```

Host (host)

For **Actions (action)** object types, specifies the TEC or SNMP host name (used for some types of actions, such as sending TEC events, or sending SNMP). If it does not apply, this field must be empty.

If you use **composer** command line to define security domains, the following syntax applies:

```
host=host_name
```

Port (port)

For **Actions (action)** object types, specifies the TEC or SNMP port number (used for some types of actions, such as sending TEC events, or sending SNMP). If it does not apply, this field must be empty.

If you use **composer** command line to define security domains, the following syntax applies:

```
port=port_number
```

Classic security model

A template file named `TWA_home/TWS/config/Security.conf` is provided with the product. During installation, a copy of the template file is installed as `TWA_home/TWS/Security.conf`, and a compiled, operational copy is installed as `TWA_home/TWS/Security`.

This version of the file contains a full access definition for the user who installed the product, `<TWS_user>`, and the system administrator (root on UNIX™ or Administrator on Windows™), who are the only users defined and allowed to connect to the user interfaces and to perform all operations on all scheduling resources.

Within the IBM Workload Scheduler network, using the security file you can make a distinction between local **root** users and the **root** user on the master domain manager by allowing local **root** users to perform operations affecting only their login workstations and providing the master domain manager **root** user the authorizations to perform operations affecting any workstation across the network.

As you continue to work with the product you might want to add more users with different roles and authorization to perform specific operations on a defined set of objects.

Do not edit the original `TWA_home/TWS/config/Security.conf` template, but follow the steps described in [Updating the security file on page 197](#) to make your modifications on the operational copy of the file.

Security management overview

The way IBM Workload Scheduler manages security is controlled by a configuration file named **security file**. This file controls activities such as:

- Linking workstations.
- Accessing command-line interface programs and the Dynamic Workload Console.
- Performing operations on scheduling objects in the database or in the plan.

In the file you specify for each user what scheduling objects the user is allowed to access, and what actions the user is allowed to perform on those objects. You can determine access by object type (for example, workstations or resources) and, within an object type, by selected attributes, such as the object's name or the workstation in the object's definition. You can use wildcards to select related sets of objects. Access rights can be granted on an "included" or an "excluded" basis, or a combination of both.

Whenever you need to change access permissions you modify the configuration file and convert it into an encrypted format (for performance and security), replacing the previous file. The system uses this encrypted *security file* from that point onwards.

Each time a user runs IBM Workload Scheduler programs, commands, and user interfaces, the product compares the name of the user with the user definitions in the *security file* to determine if the user has permission to perform those activities on the specified scheduling objects.

By default, the security on scheduling objects is managed locally on each workstation. This means that the system administrator or the `<TWS_user>` who installed the product on that system can decide which IBM Workload Scheduler users

defined on that system can access which scheduling resources in the IBM Workload Scheduler network and what actions they can perform.

Alternatively, you can centralize control of how objects are managed on each workstation. This can be configured by setting a global option. In this scenario, you configure all user permissions in the *security file* on the master domain manager. The encrypted version of the file is distributed automatically every time you run **JnextPlan**, so that all workstations have the file locally to determine the permissions of the users on that workstation.

Updating the security file

About this task

By default, every workstation in an IBM Workload Scheduler network (domain managers, fault-tolerant agents, and standard agents) has its own security file. You can maintain that file on each workstation, or, if you enable centralized security management, you can create a security file on the master domain manager and copy it to each domain manager and agent, ensuring that all IBM Workload Scheduler users are assigned the required authorization in the file (see [Centralized security management on page 201](#)). Whether working on an agent workstation for an individual security file, or on the master domain manager to modify a centralized file, the steps are just the same; all that changes are the number of users you are defining - just those on the local system or all in the IBM Workload Scheduler network.

If you are updating or upgrading your fault-tolerant agents to version 9.5 Fix Pack 2 or later, you must manually update the security file on the fault-tolerant agents in your environment to add access to folders for all of the scheduling objects that can be defined or moved into folders. These updates are especially important if you plan to use the command line on the fault-tolerant agents to perform operations on the objects in folders. More specifically, if centralized security is enabled (`enCentSec / ts = YES`), then you must first update or upgrade all of the fault-tolerant agents in your environment to version 9.5 Fix Pack 2 or later before you begin using the folder feature. If centralized security management is not enabled (`enCentSec / ts = NO`), all stanzas that reference CPU are automatically updated to include folder access, for example, `CPU CPU=@+FOLDER=" / "`, unless you use wildcard characters (`@`) as matching criteria in your stanzas, for example, `CPU CPU=@HR`. In this case, if you want to be able to move these CPUs into folders, then you need to manually update those stanzas to include access to all folders, `CPU CPU=@HR+FOLDER=" / "`.

Neither the IBM Workload Scheduler processes nor the WebSphere Application Server Liberty Base infrastructure needs to be stopped or restarted to update the security file. You just need to close any open conman user interfaces before running `makesec`.

To modify the security file, perform the following steps:

1. Configure the environment, running one of the following scripts:

In UNIX®:

- `./TWA_home/TWS/tws_env.sh` for Bourne and Korn shells
- `./TWA_home/TWS/tws_env.csh` for C shells

In Windows®:

- `TWA_home\TWS\tws_env.cmd`

2. Navigate to the following directory from where you can submit the `dumpsec` and `makesec` commands:

```
TWA_home/TWS
```

```
TWA_home\TWS
```

3. Run the `dumpsec` command to decrypt the current security file into an editable configuration file. See [dumpsec on page 198](#).
4. Modify the contents of the editable security configuration file using the syntax described in [Configuring the security file on page 202](#).
5. Close any open `conman` user interfaces using the `exit` command.
6. Stop any connectors on systems running Windows™ operating systems.
7. Run the `makesec` command to encrypt the security file and apply the modifications. See [makesec on page 199](#).
8. If you are using local security, the file will be immediately available on the workstation where it has been updated.

If you are using centralized security (see [Centralized security management on page 201](#)), perform the following steps:

- a. If you are using a backup master domain manager, copy the file to it.
- b. Distribute the centralized file manually to all fault-tolerant agents in the network (not standard, extended, or broker agents), and store it in the following directory:

For a fresh installation of version 9.5.x or later

```
TWA_DATA_DIR
```

```
TWA_home\TWS
```

Upgraded environment originating from a version earlier than 9.5:

```
TWA_home/TWS
```

```
TWA_home\TWS
```

- c. Run `JnextPlan` to distribute the Symphony file that corresponds to the new Security file.

See [dumpsec on page 198](#) and [makesec on page 199](#) for a full description of the commands.

dumpsec

Writes in an editable format the information contained in the compiled and encrypted security file. The output file can be edited and then used as input for the `makesec` command which compiles and activates the modified security settings.

Authorization

You must have `display` access to the security file and write permission in the `TWA_home/TWS` directory from where the command *must* be run.

Syntax

```
dumpsec -v | -u
```

```
dumpsec security_file [> output_file]
```

Comments

If no arguments are specified, the operational security file is sent to stdout. To create an editable copy of the security file, redirect the output of the command to an output file, using the redirect symbol.

Arguments

-v

Displays command version information only.

-u

Displays command usage information only.

security_file

Specifies the name of the security file to dump.

[> output_file]

Specifies the name of the output file, If omitted, the security file is output to the stdout.

Example

Examples

The following command dumps the operational security file (*TWA_home/TWS/Security*) to a file named **mysec**:

```
dumpsec > mysec
```

The following command dumps a security file named **sectemp** to **stdout**:

```
dumpsec sectemp
```

makesec

Compiles security definitions and installs the security file. Changes to the security file are recognized as soon as makesec has completed, or, in the case of centralized security, after **JnextPlan** has distributed it.



Note: Before running the **makesec** command, stop conman, and, on systems running Windows® operating systems, any connectors.

Authorization

You must have **modify** access to the security file and read permission in the *TWA_home/TWS* directory from where the command *must* be run.

Syntax

```
makesec -v | -u
```

```
makesec [-verify] in_file
```

Comments

The **makesec** command compiles the specified file and installs it as the operational security file (`../TWA_home/TWS/Security`). If the **-verify** argument is specified, the file is checked for correct syntax, but it is not compiled and installed.

Arguments

-v

Displays command version information only.

-u

Displays command usage information only.

-verify

Checks the syntax of the user definitions in *in_file*. The file is not compiled and installed as the security file.

in_file

Specifies the name of a file or set of files containing user definitions. Syntax checking is performed automatically when the security file is installed.

Example

Examples

Example 1: Modifying the security file definitions - full scenario

The following example shows how to modify the security file definitions:

1. An editable copy of the operational security file is created in a file named `tempsec` with the `dumpsec` command:

```
dumpsec > tempsec
```

2. The user definitions are modified with a text editor:

```
edit tempsec
```

3. The file is then compiled and installed with the **makesec** command:

```
makesec tempsec
```

Example 2: Compiling user definitions from multiple files

The following command compiles user definitions from the fileset `userdef*` and replaces the operational security file:

```
makesec userdef*
```


Centralized security management

A IBM Workload Scheduler environment where centralized security management is enabled is an environment where all workstations share the same security file information contained in the security file stored on the master domain manager and the IBM Workload Scheduler administrator on the master domain manager is the only one who can add, modify, and delete entries in the security file valid for the entire IBM Workload Scheduler environment.

This is configured with the **enCentSec** global option. By default the value assigned to the **enCentSec** option is **no**.

To set central security management, the IBM Workload Scheduler administrator must run the following steps on the master domain manager:

1. Use the **optman** command line program, to set the value assigned to the **enCentSec** global property to **yes**. For information on how to manage the global properties using **optman**, see [Setting global options on page 14](#).
2. Save the information in the security file into an editable configuration file using the **dumpsec on page 198** command.
3. Set the required authorizations for all IBM Workload Scheduler users, as described in [Configuring the security file on page 202](#)
4. Close any open **conman** user interfaces using the **exit** command.
5. Stop any connectors on systems running Windows® operating systems.
6. Compile the security file using the **makesec on page 199** command.
7. If you are using a backup master domain manager, copy the compiled security file to it as soon as possible.
8. Distribute the compiled security file to all the workstations in the environment and store it in their `TWA_home/TWS` directories.
9. Run **JnextPlan** to update the security information distributed with the `Symphony` file.

The value of the checksum of the newly compiled security file is encrypted and loaded into the `Symphony` file and distributed to all the workstations in the IBM Workload Scheduler network.

On each workstation, when a link is established or when a user connects to a user interface or attempts to issue commands on the plan, either with **conman** or the Dynamic Workload Console, IBM Workload Scheduler compares the value of the checksum in the security file delivered with the `Symphony` file with the value of the checksum of the security file stored on the workstation. If the values are equal, the operation is allowed. If the values are different, the operation fails and a security violation message is issued.

Centralized security usage notes

The following are some considerations to be aware of if centralized security management is enabled in your IBM Workload Scheduler environment.

When centralized security is enabled (**enCentSec / ts = YES**), and you plan to start using the folder feature to define or move scheduling objects into dedicated folders, then you must first update or upgrade all of the fault-tolerant agents in your environment to version 9.5 Fix Pack 2 or later. If centralized security management is not enabled (**enCentSec / ts = NO**), all stanzas that reference CPU are automatically updated to include folder access, for example, `CPU CPU=@+FOLDER= " / "`, unless you use wildcard characters (**@**) as matching criteria in your stanzas, for example, `CPU CPU=@HR`. In this case, if you want to be

able to move these CPUs into folders, then you need to manually update those stanzas to include access to all folders, `CPU`

```
CPU=@HR+FOLDER= " / " .
```

In a network with centralized security management, two workstations are unable to establish a connection if one of them has **enCentSec** turned off in its `Symphony` file or if their security file information does not match.

The only exception to the security file matching criteria introduced by the centralized security management mechanism is that a workstation must always accept incoming connections from its domain manager, regardless of the result of the security file matching process.

Centralized security does not apply to IBM Workload Scheduler operations for which the `Symphony` file is not required. Commands that do not require the `Symphony` file to run use the local security file. For example, the `parms` command, used to modify or display the local parameters database, continues to work according to the local security file, even if centralized security is active and the local security file differs from the centralized security rules.

If a workstation's security file is deleted and re-created, the checksum of the new security file will not match the value in the `Symphony` file. In addition, a run-number mechanism associated with the creation process of the `Symphony` file ensures prevention from tampering with the file.

Configuring the security file

In the security file you can 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.

When defining user authorization consider that:

- When commands are issued from the **composer** command line program, the user authorizations are checked in the security file of the master domain manager since the methods used to manage the entries in the database are invoked on the master domain manager. Therefore the user must be defined:
 - As system user on the system where the master domain manager is installed.
 - In the security file on the master domain manager with the authorizations needed to run the allowed commands on the specific objects.
- When commands are issued from the **conman** command line program, the user must be authorized to run the specific commands in the security file both on the connecting workstation and on the master domain manager where the command actually runs.

The security file is parsed one line at a time, thus any given line in the security file has been assigned a maximum length of 32768 characters. Since during the encryption process (`makesec`), one extra character is added to any string value in order to store its length, the number of "visible" characters could actually be more or less than 32768. As an example, consider the following line:

```
CPU=@+LOGON=test1, test2
```

The actual number of characters written into the encrypted Security file is determined according to this formula:

```

"CPU=" : 2 chars (token)
"@ " : 2 chars (1 + 1 for the length)
"LOGON=" : 2 chars (token)
"test1," : 7 chars (6 + 1 for the length) (string)
"test2" : 6 chars (5 + 1 for the length) (string)
-----
total : 19 chars

```

However, if counting the actual number of visible characters, there are 23 characters including the single space between test1 and test2 and the comma separating them.



Note: The "CPU" and "LOGON" each have a real length of two characters even though they actually have three and five characters respectively. This is because certain keywords are "tokenized." This can actually help reduce the apparent character count in this case.

The configuration of the security file is described in these sections:

- [Security file syntax on page 203](#)
- [Specifying user attributes on page 205](#)
- [Specifying object types on page 211](#)
- [Specifying object attributes on page 213](#)
- [Specifying access on page 219](#)
- [The <TWS_user> - special security file considerations on page 241](#)

Security file syntax

The syntax of the security file is as follows:

Syntax

```
[# comment]
```

```
user definition_name user_attributes
```

```
begin [* comment]
```

```
object_type [object_attributes]. access[=keyword[,keyword]...]
```

```
[object_type [object_attributes]. access[=keyword[,keyword]...] ]...
```

```
end | continue
```

Arguments

```
[# | *] comment
```

All text following a pound sign or an asterisk and at least one space is treated as a comment. Comments are not copied into the operational security file installed by the **makesec** command.

user *definition_name*

Specifies the name of the user definition. The name can contain up to 36 alphanumeric characters and must start with an alphabetic character.

user *attributes*

Contains one or more attributes that identify the user or users to whom the definition applies. For details of how to define user attributes, see [Specifying user attributes on page 205](#).

begin

Begins the part containing object statements and accesses within the user definition.

object *type*

Identifies the type of object (for example: workstation, resource, or prompt) to which access is to be given for the specified user or users. All object types that the specified user or users needs to access must be explicitly defined. If they are not, no access will be given. For details of how to define object types, see [Specifying object types on page 211](#).

object *attributes*

Contains one or more attributes that identify the specific objects of the defined object type to which the same access is to be given. If no object attributes are defined, access is given to all objects of the defined object type. For details of how to define object attributes, see [Specifying object attributes on page 213](#).

access[=*keyword*[,*keyword*]...]

Describes the access to the specified objects given to the selected users. If none is specified (by specifying just the keyword "access") no access is given to the associated objects. If **access=@** then all access rights are assigned to the specified users. For details of how to define access, see [Specifying access on page 219](#).

continue

Allows a user to inherit authorization from multiple *stanzas*. Add the `Continue` keyword before the `Begin` keyword of each subsequent *stanza* to request that IBM Workload Scheduler must not stop at the first *stanza*, but must continue including also the following *stanzas* that match the user definition. The user gets the accesses for the first matching entry of each *stanza*. For an example of the use of the `Continue` keyword, see [Users logged into multiple groups \[continue keyword\] on page 247](#).

end

Terminates the user definition. The users defined in the user definition that terminates with an `end` statement do not match any subsequent user definition.

Wildcards

The following wildcard characters are permitted in user definition syntax:

?

Replaces one alphanumeric character.

@

Replaces zero or more alphanumeric characters.

For information about variables supplied with the product that can be used in object attributes, refer to [Using variables in object attribute definitions on page 219](#). Refer to [Sample security file on page 242](#) for an example on how to use variables.

Specifying user attributes

The user attributes define *who* has the access that is going to be subsequently defined. They can identify one user, a selection of users, a group of users, a selection of groups of users, or all users. You can also exclude one or more specific users or groups from a selection. As well as being identified by logon ID and group name, users can also be described by the workstation from which they log on. And finally, you can mix selection criteria, for example selecting all users in a named group that can access from a set of workstations identified by a wildcard, but excluding a specific set of users identified by their logon IDs.

A user must be uniquely identified. If different users have the same identifier, an error is issued when `makesec` command is run. You must edit the security file by using `dumpsec` command, assign a unique identifier to users, and rerun the `makesec` command.

The general syntax

You make this selection by specifying one or more user attributes. Each user attribute is specified as follows:

user_attribute_type=value

user_attribute_type

Can be *cpu* (workstation), *group*, or *logon*

value

Identifies an individual *cpu* (workstation), *group*, or *logon*, or, by using wildcards, can identify a set of any of these.

Including or excluding

Each attribute can be *included* or *excluded* from the selection.

Thus, for each *attribute type*, your options are one of the following:

Include all

This is the default. Thus, for example, if you want to include all *groups*, you need add no user attribute with respect to any group.

Include a selection

This can be defined in one of these ways:

- By specifically including users you want to select (individuals or one or more sets)
- By specifically excluding (from the *include all* default) all users you do *not* want to select
- By specifically including a set of users and then excluding some of those contained in the set

Which of these options you choose is determined by which is easier to specify.

Using the include or exclude symbols

Include

Precede the user attribute expression by a plus (+) sign. All users identified by the expression will be selected, unless they are also selected by an *exclude* expression. If the first attribute in your definition is an *include*, it does not need to have a (+) sign defined, because the sign is implicit.

The default (if you specify no user attributes) is to include all users, on all workstations, in all groups, so if you want to define, for example, all users except one named user, you would just supply the *exclude* definition for the one user.

Exclude

Precede the user attribute expression by a tilde (~) sign. All users identified by the expression will *never* be selected, regardless of if they are identified by any *include* expressions.

Selection expressions

You can use the following different types of selection expression:

Basis selection expressions

Include only one attribute

```
user_attribute_type=value
```

For example, to include one named user logon ID, and exclude all other users:

```
logon=jsmith1
```

Exclude one attribute

```
~user_attribute_type=value
```

For example, to exclude one set of logon IDs identified by a wildcard (those that start with the letter "j"), but include all others:

```
~logon=j*
```

Include only several attributes of the same type

```
user_attribute_type=value[,value]...
```

For example, to include three specific users and exclude all others:

```
logon=jsmith1,jbrown1,jjones1
```

Exclude several attributes of the same type

```
~user_attribute_type=value[,value]...
```

For example, to exclude three specific users and include all others:

```
~logon=jsmith1,jbrown1,jjones1
```

Complex selection expressions**Include users identified by different selection expressions**

```
basic_selection_expression[+basic_selection_expression]...
```

The selection expressions can be of the same or a different attribute type:

Same attribute type

An example of the same attribute type is the following, which selects all the groups beginning with the letter "j", as well as those with the letter "z":

```
group=j@+group=z@
```

If the first selection identifies 200 users, and the second 300, the total users selected is 500.

Different attribute type

An example of selection expressions of a different attribute type is the following, which selects all the groups beginning with the letter "j", as well as all users with IDs beginning with a "6":

```
group=j@+logon=6@
```

If the first selection identifies 200 users, and the second 20, of whom 5 are also in the first group, the total users selected is 5.

Exclude users identified in one selection expressions from those identified in another

```
basic_selection_expression[~basic_selection_expression]...
```

Same attribute type

The selection expressions can be of the same attribute type, provided that the second is a subset of the first. An example of the same attribute type is the following, which selects all the workstations beginning with the letter "j", but excludes those with a "z" as a second letter:

```
group=j@~group=jz@
```

If the first selection identifies 200 users, and the second 20, the total users selected is 180. Note that if the second expression had not been a subset of the first, the second expression would have been ignored.

Different attribute type

Selection expressions of a different attribute type do not have to have a subset relationship, an example being the following, which selects the group "mygroup", but excludes from the selection all users in the group with IDs beginning with a "6":

```
group=mygroup~logon=6@
```

If the first selection identifies 200 users, and the second 20, of whom 5 are also in the first group, the total users selected is 195.

Multiple includes and excludes

You can link together as many include and exclude expressions as you need to identify the precise subset of users who require the same access. The overall syntax is thus:

```
[~]user_attribute_type=value[,value]... [{+|~}user_attribute_type=value[,value]...
```



Note: Making your *first* user attribute an *exclude* means that *all* user attributes of that type are selected *except* the indicated *value*. Thus, `~user_attribute_type=value` equates to the following:

```
user_attribute_type=@~same_user_attribute_type=value
```

However, if you use this syntax, you cannot, and do not need to, specifically add `+user_attribute_type=@`, after the negated item, so you do not define:

```
~user_attribute_type=value+same_user_attribute_type=@
```

Order of user definition

You must order user definitions from most specific to least specific. IBM Workload Scheduler scans the security file from top-down, with each user ID being tested against each definition in turn. If the user ID is satisfied by the definition, it is selected, and the matching stops.

For example:

Incorrect:

```
#First User Definition in the Security File
USER TwsUser
CPU=@+LOGON=<TWS_user>
Begin
job name=@ access=modify
End

#Second User Definition in the Security File
USER Twsdomain:TwsUser
CPU=@+LOGON=TWSDomain\\<TWS_user>
Begin
job name=@ access=display
End
```


The definitions are intended to determine the following:

1. Users on all workstations with a logon of "TWS_user" will be given "modify" access to all jobs
2. Users on all workstations with a logon of "TWSDomain\TWS_user" will be given "display" access to all jobs

However, all users with a logon of "TWS_user" will satisfy the first rule, regardless of their domain, and will be given "modify" access to all jobs. This is because defining a user without its domain is a shorthand way of defining that user ID in *any* domain; it is the equivalent of "@\TWS_User". So the second rule will never be satisfied, for any user, because the matching for the "TWS_user" stops after a successful match is made.

Correct

```
#First User Definition in the Security File
USER Twsdomain:Tws_User
CPU=@+LOGON="TWSDomain\<TWS_user>"
Begin
job name=@ access=display
End

#Second User Definition in the Security File
USER Tws_User
CPU=@+LOGON=<TWS_user>
Begin
job name=@ access=modify
End
```

By putting the more specific definition first, both object access definitions are applied correctly.

See [Sample security file on page 242](#) for a practical example.

User attribute types - detailed description

The *user_attribute_types* and their associated *values* can be any of the following:

cpu={*[folder]/workstation*}/@/{*[folder]/workstation*}/@/}

where:

workstation

Specifies the workstation on which the user is logged in. Wildcard characters are permitted. The following IBM Workload Scheduler variables can be used:

\$master

Means that the user is logged in on the IBM Workload Scheduler master domain manager.

\$manager

Means that the user is logged in on the IBM Workload Scheduler domain manager.

\$thiscpu

Means that the user is logged in on the IBM Workload Scheduler workstation on which the security check is running.

@

Specifies that the user is accessing IBM Workload Scheduler with the Dynamic Workload Console, or is logged in on any IBM Workload Scheduler workstation.

group=groupname

Specifies the name of the group of which the user is a member. Available for both UNIX™ and Windows™ users. Wildcard characters are permitted.

logon={user name@}

where:

user name

Specifies the user ID with which the user is logged in on a IBM Workload Scheduler workstation. Wildcard characters are permitted. The **cpu=** attribute must be set to a specific workstation name (no wildcards) or @.

The `user name` value can have one of the following formats:

user name

The Windows user. For example if you use the `user1` value in the `logon` field, in the `Security` file you have the following line:

```
.....
logon=user1
.....
```

domain\user name

The user belongs to a Windows domain. Insert the escape character `\` before the `\` character in the `domain\user name` value. For example if you use the `MYDOMAIN\user1` value in the `logon` field, in the `Security` file you have the following line:

```
.....
logon=MYDOMAIN\user1
.....
```

user name@internet_domain

The user belongs to an internet domain. The user name is in User Principal Name (UPN) format. UPN format is the name of a system user in an email address format. The user name is followed by the "at sign" followed by the name of the Internet domain with which the user is associated.

Insert the escape character '\' before the '@' character in the `user` `name@internet_domain` value. For example if you use the `administrator@bvt.com` value in the `logon` field, in the `Security` file you have the following line:

```
.....
logon=administrator\@bvt_env.com
.....
```

For more information about the use of the wildcard with the `domain\user` name and `user` `name@internet_domain` format in the `Security` file, see [Sample security file on page 242](#).



Note:

1. If the WebSphere Application Server Liberty Base security configuration option **useDomainQualifiedUserNames** is set to *true*, each user ID defined in the security file must have the format `domain\username` to use the product from one of the following:

- **composer**
- **Dynamic Workload Console**
- **logman**
- **optman**
- **planman**

For more information on WebSphere Application Server Liberty Base security configuration, see [Changing the security settings on page 424](#).

2. If the user is defined on a Windows™ 2003 system, or when upgrading the Windows™ operating system from an older version to one of those mentioned above, make sure you add the **Impersonate a client after authentication** right to the user settings.

@

Specifies any user logged in with any name or being a member of any IBM administrators group.

Specifying object types

Specify one or more object types that the user or users in the associated user definition is authorized to access. If you specify the object type but no attributes, the authorized actions defined for the user with the **access** keyword apply to all the objects of that type defined in the IBM Workload Scheduler domain. If an object type from the following list is omitted for a user or users, no objects of that type can be accessed.

The object types are the following:

action

Actions defined in scheduling event rules

calendars

User calendars

cpu

Workstations, domains and workstation classes

event

Event conditions in scheduling event rules

eventrule

Scheduling event rule definitions

file

IBM Workload Scheduler database file

folder

The folder within which scheduling objects such as, jobs, job streams, and workstations, to name a few, are defined.

job

Scheduled jobs and job definitions

parameter

Local parameters. See [note on page 213](#) below.

prompt

Global prompts

report

The reports on the Dynamic Workload Console that have the following *names*:

RUNHIST

Job Run History

RUNSTATS

Job Run Statistics

WWS

Workstation Workload Summary

WWR

Workstation Workload Runtimes

SQL

Custom SQL

ACTPROD

Actual production details (for current and archived plans)

PLAPROD

Planned production details (for trial and forecast plans)

Permission to use these reports is granted by default to the `<TWS_user>` on fresh installations.

resource

Scheduling resources

runcygrp

Run cycle groups

schedule

Job streams

userobj

User objects

vartable

Variable tables. This includes authorization to the variable definitions in the tables. See the [note on page 213](#) below.

wkldappl

Workload applications



Note: Starting from version 8.5, the **parameter** object type is reserved for parameters created and managed in a local parameter database with the `parms` utility command, while authorization to act on global variables is managed using the **vartable** object type. For this reason, when the security file is migrated from previous versions to 8.5, a `vartable` security definition for the default variable table is added to match each `parameter` definition found, as part of the upgrade process documented in the *IBM Workload Scheduler: Planning and Installation Guide*.

Specifying object attributes

Specify one or more attributes that identify a set of objects that the user of the user definition is authorized to access. If you specify the object type but no object sets, the authorized actions defined for the user with the **access** keyword apply to all the objects of that type defined in the IBM Workload Scheduler domain.

The general syntax

Each object attribute is specified as follows:

```
object_attribute=value
```

object_attribute

Object attributes differ according to the object. All objects can be selected by *name*, but some, *jobs*, for example, can be selected by the *workstation* on which they run. See [Object attribute on page 214](#) for full details of which attributes are available for each object type.

value

Identifies an individual object, or, by using wildcards, a set of objects. See [Specifying object attributes on page 213](#) for full details of which attributes are available for each object type.

Object attribute

[Specifying object attributes on page 213](#) lists object attributes that are used to identify a specific set of objects from all objects of the same type. For example, access can be restricted to a set of resource objects having the same name or being defined on the same workstation, or both.

Table 40. Object attribute types for each object type

Attribute Object	name	cpu	fol der	cpuf older	cus tom	jcl	jcltype	logon	provider	type	host	port
action			✓						✓	✓	✓	✓
calendar	✓		✓									
cpu (workstation)		✓		✓						✓		
event					✓				✓	✓		
eventrule	✓		✓									
file	✓											
folder	✓											
job	✓	✓	✓	✓		✓	✓	✓				
lob	✓											
parameter	✓	✓	✓	✓								
prompt	✓		✓									
report	✓											
resource	✓	✓	✓	✓								
runcygrp	✓		✓									
schedule (job stream)	✓	✓	✓	✓								
userobj		✓		✓				✓				
vartable	✓		✓									

Table 40. Object attribute types for each object type (continued)

Attribute	name	cpu	fol der	cpuf older	cus tom	jcl	jcltype	logon	provider	type	host	port
Object												
wkldappl	✓		✓									

**Note:**

- Granting access to a workstation class or a domain means to give access just to the object itself, and grant no access to the workstations in the object.
- When you specify access rights on a folder, the access rights apply also to all sub-folders.

Including or excluding

Each attribute can be *included* or *excluded* from the selection using the plus (+) and tilde (~) symbols, in the same way as for the user attributes.

Selection expressions

The detailed syntax and use of the selection expressions for objects is the same as that used to select users:

```
[~]object_attribute=value[,value]...[+|~]object_attribute=value[,value]...
```

Order of object definition

You must order object definitions from most specific to least specific, in the same way as for user attributes. For example,

Incorrect

```
job name=@ access=display
job name=ar@ access=@
```

In this case, a job with the name beginning with "ar" would satisfy the first definition, and so would be given the display access, not all access.

Correct

```
job name=ar@ access=@
job name=@ access=display
```

Ensure that you order object definitions from most specific to least specific also when you use the `Continue` keyword.

The `Continue` keyword allows a user to inherit authorization from multiple *stanzas*. The user receives accesses as defined in the first matching entry of each *stanza* that matches the user definition. For an example of a security file with the `Continue` keyword, see [Users logged into multiple groups \[continue keyword\] on page 247](#)

Specifying object attribute values

The following describes the values allowed for each object attribute type:

name=*name*[,*name*]...

Specifies one or more names for the object type. Wildcard characters are permitted. Multiple names must be separated by commas.

- The following values apply to the file object type:

globalopts

Allows the user to set global options with the `optman` command. Gives the following access types:

- Display access for `optman ls` and `optman show`
- Modify access for `optman chg`

prodsked

Allows the user to create, extend, or reset the production plan.

security

Allows the user to manage the security file.

Symphony

Allows the user to run **stageman** and **JnextPlan**.

trialsked

Allows the user to create trial and forecast plans or to extend trial plans.



Note: Users who have restricted access to files should be given at least the following privilege to be able to display other objects (ie. calendars and cpus):

```
file name=globalopts access=display
```

- For the event object type use one or more of the event type names listed in the `TWSObjectsMonitor events` table or the `FileMonitor events` table in the *IBM Workload Scheduler: User's Guide and Reference*.
- For the action object type use one or more of the action type names listed in the table *Action types by action provider* in the *IBM Workload Scheduler: User's Guide and Reference*.
- For the **var**table object type, you can use the \$DEFAULT value for the **name** attribute to indicate the default variable table. This selects the table defined with the `isdefault` attribute.

cpu=*workstation* + **folder=***foldername*

Specifies one or more workstation, domain, or workstation class names. Workstations and workstation classes can optionally be defined in a folder; if defined, the folder can be specified in the folder attribute. Wildcard characters are permitted. Multiple names must be separated by commas. If this attribute is not specified, all

defined workstations and domains can be accessed. Workstation variables can be used - see [Using variables in object attribute definitions on page 219](#).

folder=foldername

Scheduling objects such as, jobs, job streams, and workstations, to name a few, can be defined in a folder. A folder can contain one or more scheduling objects, while each object can be associated to only one folder. The default folder is the root folder (/).

cpufolder=foldername

The folder within which the workstation or workstation class is defined.

custom=value[,value]...

Use this attribute to assign access rights to events defined in event plug-ins. The precise syntax of the value will depend on the plug-in. For example:

- Specify different rights for different users based on SAP R/3 event names when defining event rules for SAP R/3 events.
- Define your own security attribute for your custom-made event providers.
- Specify the type of event that is to be monitored. Every event can be referred to an event provider.

jcl="path" | "command" | "jsdl"

Specifies the command or the path name of a job object's executable file. The command or path must be enclosed in double quotation marks (" "). Wildcard characters are permitted. If omitted, all defined job files and commands qualify.

You can also specify a string contained in the task string of a JSDL definition to be used for pattern matching. Ensure that the string begins and ends with the @ wildcard character and that it is entirely enclosed in double quotation marks as follows: "@my_string>@".

jcltype=[scriptname | docommand]

Specifies that the user is allowed to act on the definitions of jobs that run only scripts (if set to scriptname) or commands (if set to docommand). Use this optional attribute to restrict user authorization to actions on the definitions of jobs of one type or the other only. Actions are granted for both scripts and commands when jcltype is missing.

A user who is not granted authorization to work on job definitions that run either a command or a script is returned a security error message when attempting to run an action on them.

logon=username[,...]

Specifies the user IDs. Wildcard characters are permitted. Multiple names must be separated by commas. If omitted, all user IDs qualify.

The user ID can be a Windows domain user or an internet domain user and must be defined in one of the following formats:

domain\user name

The user belongs to a Windows domain. Insert the escape character '\ ' before the '\ ' character in the `domain\user name` value. For example if you use the `MYDOMAIN\user1` value in the logon field, in the `Security` file you have the following line:

```
.....
logon=MYDOMAIN\user1
.....
```

user name@internet_domain

The user belongs to an internet domain. The user name is in User Principal Name (UPN) format. UPN format is the name of a system user in an email address format. The user name is followed by the "at sign" followed by the name of the Internet domain with which the user is associated.

Insert the escape character '\ ' before the '@' character in the `user name@internet_domain` value. For example if you use the `administrator@bvt.com` value in the logon field, in the `Security` file you have the following line:

```
.....
logon=administrator\bvt_env.com
.....
```

provider=provider_name[,...]

For **action** object types, specifies the name of the action provider.

For **event** object types, specifies the name of the event provider.

Wildcard characters are permitted. Multiple names must be separated by commas. If `provider` is not specified, no defined objects can be accessed.

type=type[,...]

For **action** object types, is the `actionType`.

For **event** object types, is the `eventType`.

For **cpu** object types, the permitted values are those used in composer or the Dynamic Workload Console when defining workstations, such as `manager`, `broker`, `fta`, `agent`, `s-agent`, `x-agent`, `rem-eng`, `pool`, `d-pool`, `cpuclass`, and `domain`.



Note: The value `master`, used in `conman` is mapped against the `manager` security attributes.

Wildcard characters are permitted. Multiple names must be separated by commas. If **type** is not specified, all defined objects are accessed for the specified providers (this is always the case after installation or upgrade, as the type attribute is not supplied by default).

host=host_name

For **action** object types, specifies the TEC or SNMP host name (used for some types of actions, such as sending TEC events, or sending SNMP). If it does not apply, this field must be empty.

port=port_number

For **action** object types, specifies the TEC or SNMP port number (used for some types of actions, such as sending TEC events, or sending SNMP). If it does not apply, this field must be empty.

Using variables in object attribute definitions

The following variables supplied with the product can be used in object attributes:

Workstation identifiers

\$master

The IBM Workload Scheduler master domain manager.

\$manager

The IBM Workload Scheduler domain manager.

\$thiscpu

The workstation on which the user is running the IBM Workload Scheduler command or program.

Variable table identifiers

\$default

The name of the current default variable table.

Specifying access

About this task

Specify the type of access the selected users are allowed to have to the specified objects as follows:

access[=*keyword*[,*keyword*]...]

- To specify that no actions are permitted, use **access=**
- To specify that all actions are permitted, use **access=@**
- To specify any other access, consult the access tables, by object type, below.

How the access tables are organized

The access tables for object types are as follows:

Object types - calendar, cpu, eventrule, folder, job, prompt, resource, run cycle group, schedule, userobj, vartable - using in composer on page 221

Most of the **composer** and GUI database maintenance actions are common to most objects, so they are listed in a table of common object access keywords.

Object type - action on page 224

This gives the access rights for action objects, which are not included in the common table.

Object type - calendar on page 225

This gives the access rights for calendars, which are different or additional to those in the common table.

Object type - cpu on page 225

This gives the access rights for workstations (cpus), which are different or additional to those in the common table.

Object type - event on page 227

This gives the access rights for events, which are different or additional to those in the common table.

Object type - file on page 228

This gives the access rights for files, which are different or additional to those in the common table.

Object type - folder on page 229

This gives the access rights for folders, which are different or in addition to those in the common table.

Object type - job on page 231

This gives the access rights for jobs, which are different or additional to those in the common table.

Object type - parameter on page 235

This gives the access rights for local parameters, which are not included in the common table.

Object type - prompt on page 235

This gives the access rights for prompts, which are different or additional to those in the common table.

Object type - report on page 236

This gives the access rights for reports, which are different or additional to those in the common table.

Object type - resource on page 237

This gives the access rights for resources, which are different or additional to those in the common table.

Object type - run cycle group on page 238

This gives the access rights for run cycle groups, which are different or additional to those in the common table.

Object type - schedule on page 238

This gives the access rights for job streams (schedules), which are different or additional to those in the common table.

Object type - userobj on page 239

This gives the access rights for userobj, which are different or additional to those in the common table.

Object type - vartable on page 240

This gives the access rights for variable tables, which are not included in the common table.

Object type - workload application on page 241

This gives the access rights for workload applications, which are not included in the common table.

Object types - calendar, cpu, eventrule, folder, job, prompt, resource, run cycle group, schedule, userobj, vartable - using in composer

The following table gives the access keywords required to use composer to work with objects of the following types:

- calendar
- cpu
- eventrule
- folder
- job
- prompt
- resource
- run cycle group
- schedule
- userobj
- vartable

**Note:**

- The `parameter` keyword is reserved for parameters created and managed in a local parameter database with the `parms` utility command.

For more information about `parms`, see the related section in the *User's Guide and Reference*.

- If you plan to upgrade your environment from a previous version of IBM® Workload Scheduler and use event-driven workload automation, you need to manually add the `display` access keyword to all workstations on which you plan to define File Monitor events.

For more information about event-driven workload automation, see the related section in the *User's Guide and Reference*.

Table 41. Access keywords for composer actions

		Activity	Access keywords required
Composer	add	Add new object definitions in the database from a file of object definitions. Unlock access is needed to use the <code>runlock</code> attribute. For variable tables, to <i>add</i> individual variable entries within a table, the table must have <i>modify</i> access.	add, modify, unlock

Table 41. Access keywords for composer actions (continued)

	Activity	Access keywords required
	To add a CPU object as a member of a workstation class, you must add <i>use</i> access to the CPU object.	
add event rule	Add an event rule of type File Monitor.	display
create	Create a text file of object definitions in the database. Modify access is need to use the <code>:lock</code> attribute. For variable tables, create individual variable entries within the table.	display, modify
delete	Delete object definitions from the database. For variable tables, to <i>delete</i> individual variable entries within a table, the table must have <i>modify</i> access.	delete
display	Display object definitions in the database.	display
extract	Extract a text file of object definitions from the database.	display
list	List object definitions in the database.	If the enListSecChk global option is set to <i>yes</i> on the master domain manager then, either <i>list</i> , or <i>list</i> and <i>display</i> are required.
lock	Lock object definitions in the database.	modify
modify	Modify object definitions in the database. Definitions are extracted into a file. After you have edited the file the definitions are used to replace the existing ones. For variable tables, to <i>modify</i> individual variable entries within a table, the table must have <i>modify</i> access.	add, modify
new	Create object definitions in the database from a template.	add, modify
print	Print object definitions in the database.	display
rename	Rename object definitions in the database. You need add access to the new object and delete and display access to the old object.	add, delete, display

Table 41. Access keywords for composer actions (continued)

		Activity	Access keywords required
	replace	Replace object definitions in the database. Unlock access is needed to use the <code>runlock</code> attribute.	add, modify, unlock
	unlock	Unlock object definitions in the database. For variable tables, unlocking a table unlocks all the variables contained therein. Unlocking a variable unlocks the entire table where it is defined.	unlock
Dynamic Workload Console	Add event rule	Add an event rule of type File Monitor.	display
	Create object in database	Add new object definitions in the database.	add
	Delete object in database	Delete object definitions from the database. Unlock access is needed to use the <code>runlock</code> option.	delete
	Display object in database	Display object definitions in the database.	display
	List object in database	List object definitions in the database.	display
	Modify object in database	Modify object definitions in the database. Unlock access is needed to use the <code>runlock</code> option.	modify
	Unlock object in database	Unlock object definitions in the database locked by another user.	unlock
	Perform operations for job types with advanced options, both those supplied with the product and the additional types implemented through the custom plug-ins. You can define and perform operations on job types with advanced options with the Workload Designer.	Perform operations for job types with advanced options in the database.	run

Table 41. Access keywords for composer actions (continued)

	Activity	Access keywords required
Using the workload service assurance feature	All activities For any user to perform any workload service assurance activities, the <code><TWS_user></code> must have the following access keywords for all <code>cpu</code> , <code>job</code> , and <code>schedule</code> objects:	display, modify, list

Example

To allow a user to use the composer list, display, and modify actions on event rules, specify:

```
eventrule      access=add,display,modify
```

Object type - action

The following table gives the access keywords required for actions:

Table 42. Actions - access keywords

	Activity	Access keywords required
Dynamic Workload Console	Display action instances	display
	List action instances.	list
Dynamic Workload Console	Use these specific action types in event rule definitions.	use

conman

- For actions with provider `TWSAction` and types `sbj`, `sbd`, or `sbs`, you must set this keyword in combination with the `submit` access keyword for the specific jobs and job streams specified in the action.
- For actions with provider `TWSAction` and type `reply`, you must set this keyword in combination with the `reply` access keyword set for the specific prompts specified in the action.

The `<TWS_user>` of the workstation running the event processing server must have these `submit` and `reply` authorizations, otherwise the event processing server will not be able to run this type of actions.

Example

To allow a user to use the Dynamic Workload Console to list action instances, specify:

```
action      access=list
```


Object type - calendar

The following table gives the additional access keywords required to work with calendars, other than those described in [Table 41: Access keywords for composer actions on page 221](#):

Table 43. Calendar - additional access keywords

		Activity	Access keywords required
Composer	Use calendars in:		use
Dynamic Workload Console	<ul style="list-style-type: none"> • job streams • run cycles • run cycle groups 		

Example 1

To allow a user to only use calendars when working with job streams in any of the interfaces, specify:

```
calendar      access=use
```

Example 2

To allow a user to display, list, and print calendars, and use them when working with job streams in any of the interfaces, specify:

```
calendar      access=display,use,list
```

Object type - cpu

The following table gives the additional access keywords required to work with cpus (includes workstations, domains, and workstation classes), other than those described in [Table 41: Access keywords for composer actions on page 221](#):

Table 44. Cpus - additional access keywords

		Activity	Access keywords required
Conman	console	View and send messages to the IBM Workload Scheduler conman console.	console
Dynamic Workload Console	deployconf	Force update the monitoring configuration file for the event monitoring engine.	start
	fence	Alter workstation job fences in the production plan.	fence

Table 44. Cpus - additional access keywords (continued)

	Activity	Access keywords required	
	limit cpu	Alter workstation job limits in the production plan.	limit
	link	Open workstation links.	link
	resetfta	Generates an updated Sinfonia file and sends it to a fault-tolerant agent on which the Symphony file has corrupted.	resetfta
	showcpus	Display workstations, domains and links in the plan.	list
	shutdown	Shut down IBM Workload Scheduler processing.	shutdown
	start	Start IBM Workload Scheduler processing.	start
	startappserver	Start the application server.	start
	starteventprocessor	Start the event processor server.	start
	startmon	Start the event monitoring engine.	start
	stop	Stop IBM Workload Scheduler processing.	stop
	stop;progressive	Stop IBM Workload Scheduler processing progressively.	stop
	stopappserver	Stop the application server.	stop
	stopeventprocessor	Stop the event processor server.	stop
	stopmon	Stop the event monitoring engine.	stop
	switcheventprocessor	Switch the event processor server from the master domain manager to the backup master domain manager or vice versa.	start, stop
	switchmgr	Switch the domain manager functionality to a workstation.	start, stop
	unlink	Close workstation links.	unlink
	upgrade	Install a fix pack or upgrade to a later version fault-tolerant agents and dynamic agents.	manage
Startup	Start IBM Workload Scheduler processing.		start
Using the workload service assurance feature	All activities	For any user to perform any workload service assurance activities, the <TWS_user> must have the following access keywords:	display, modify, list

Table 44. Cpus - additional access keywords (continued)

	Activity	Access keywords required
Submit a job	When submitting a job defined in a folder, <code>use</code> access is required on the workstation (<code>cpu</code>) where the job is defined, in addition to access to the folder and the objects it contains.	<code>use</code>
Submit a job stream	When submitting a job stream defined in a folder, <code>use</code> access is required on the workstation (<code>cpu</code>) where the job is defined, in addition to access to the folder and the objects it contains.	
Composer	Use a File Monitor event on the workstation where the file resides.	<code>display</code>
Dynamic Workload		
Console		



Note: If you plan to upgrade your environment from a previous version of IBM® Workload Scheduler and use event-driven workload automation, you need to manually add the `display` access keyword to all workstations on which you plan to define File Monitor events.

For more information about event-driven workload automation, see the related section in the *User's Guide and Reference*.

Example

To allow a user to display, list, and print workstation, workstation class, and domain definitions, link and unlink workstations, and access all workstations defined in the root (`/`) folder, specify:

```
cpu name = @ + folder = / access=display,link,unlink
```

Object type - event

The following table gives the access keywords required to work with events:

Table 45. Events - access keywords

	Activity	Access keywords required
Composer	Use an event in an event rule definition.	<code>use</code>
Dynamic Workload		
Console		



Note: If you plan to upgrade your environment from a previous version of IBM® Workload Scheduler and use event-driven workload automation, you need to manually add the `display` access keyword to all workstations on which you plan to define File Monitor events.

For more information about event-driven workload automation, see the related section in the *User's Guide and Reference*.

Example

To allow a user to use an event in an event rule definition, specify:

```
event          access=use
```

Object type - file

The following table gives the access keywords required to work with files (valid only for the command line).

You must specify the [file names on page 216](#) to which the type of access applies.

Table 46. Files - access keywords

	Activity	Access keywords required
dumpsec	Create a text file of the settings contained in the compiled security file.	display
JnextPlan	Generate the production plan.	build
makesec	Compile the security file from a text file of the settings.	modify
optman	ls List all global options.	display
	show Show the details of a global option.	display
	change Change the details of a global option.	modify
planman	deploy Manually deploy event rules.	build
prodsked	Work with the production plan.	build
stageman	Carry forward incompletd job streams, archive the old production plan, and install the new production plan.	build

Example 1

To allow a user to manage the `globalopts` file, specify:

```
file name=globalopts access=display,modify
```

Example 2

To allow a user to run **JnextPlan**, specify:

```
file          access=build
```



Note: The user will also be able to run **planman deploy**, **prodsked**, and **stageman**.

Object type - folder

The following table gives the additional access keywords required to work with folders, in addition to those common to most objects described in [Table 41: Access keywords for composer actions on page 221](#):

Table 47. folders - access keywords

		Activity	Access keywords required
Composer	chfolder	Change the current folder or working directory.	display
	listfolder	Lists folders defined in the database.	list, or list and display
	mkfolder	Creates a new folder definition in the database.	add
	rmfolder	Deletes folders defined in the database.	delete
	renamefolder	Renames a folder definition in the database.	delete access to the folder with the old name, and add access to the folder with the new name
Conman	Chfolder	Changes the working directory or current directory.	display
	Listfolder	Lists folders defined in the plan.	list, or list and display

See [Example on page 229](#) for detailed examples about how to restrict access to folders.

For more information about designing workflow folders, see the related section in *Dynamic Workload Console User's Guide*.

Example

The following examples demonstrate how to restrict access to specific folders. Even with access to a folder, a user still needs additional rights to work with the objects defined in it. When submitting a job or job stream defined in a folder, `use` access is required on the workstation (cpu) where the job is defined, in addition to access to the folder and the objects it contains.

IBM® Workload Scheduler administrator can grant administrator permissions to a user on a folder, `ACL`, so that the user can freely assign access control lists to other users on the same folder or any sub-folders. Users can then access the objects in the folder or sub-folders. For more information about delegating administrator access to users and groups on a folder, see the related topic in the *Administration Guide*.

Examples

Tim the IBM® Workload Scheduler administrator, delegates Linda, the `app1_admin` user, permissions on the folder `/PRD/APP1` and any sub-folders, by assigning her the `ACL` access on the folder. With this access, Linda can create access control lists to grant access to the folder or sub-folders to other users with a predefined role. The following is the security file for Linda, the `app1_admin` user:

```
#####
#   Sample Security File
#####
USER APPADMINofPRDAPP1  cpu=JUPITER+LOGON=app1_admin
begin
# OBJECT      ATTRIBUTES                                ACCESS CAPABILITIES
# -----
job           cpu=JUPITER  + folder = "/PRD/APP1","/PRD/APP1/"
              access=add,delete,display,modify,use,list,unlock
schedule     cpu=JUPITER  + folder = "/PRD/APP1","/PRD/APP1/"
              access=add,delete,display,modify,use,list,unlock
folder       name="/PRD/APP1","/PRD/APP1/"
              access=add,delete,display,modify,use,list,unlock,ac1
```

User `jsmith` is granted unrestricted access to jobs and job streams defined in the folder named `APPS` and on the workstation named `JUPITER`, specify:

```
#####
#   Sample Security File
#####
user jsmith  cpu=JUPITER
begin
# OBJECT      ATTRIBUTES                                ACCESS CAPABILITIES
# -----
job           cpu=JUPITER  + folder = /APPS/  access=@
schedule     cpu=JUPITER  + folder = /APPS/  access=@
cpu          cpu=JUPITER+LOGON=jsmith                access=use
folder       name=/APPS/                             access=add,delete,display,
              modify,use,list,unlock,ac1
```

To allow a user to have the specified rights on any folder, the root folder and any sub-folders, specify:

```
folder  name=/      access=add,delete,display,modify,use,list,unlock
```

To grant a user access only to the root folder (`/`), you can omit specifying the folder object in the security file. This is the same behavior as in security files for releases prior to Version 9.5. After upgrading to Version 9.5, all of the objects are moved to the root folder, so if you continue to use your old security file which does not include the `v95fp1` attribute or object (for example, for jobs, `JOB CPU=@ ACCESS=ADD,ADDDEP,...,RERUN,SUBMIT,USE,LIST,UNLOCK`, then users have access to only the root (`/`) folder by default.

To allow a user to have the specified rights only on the "APPS" folder, specify:

```
folder name=/APPS/ access=add,delete,display,modify,use,list,unlock
```

To allow a user to have the specified rights on the folder "APPS" and its sub-folders, specify:

```
folder name=/APPS access=add,delete,display,modify,use,list,unlock
```

To allow a user to have the specified rights only on folder "APP1" and its sub-folders, specify:

```
folder name=/APPS/APP1 access=add,delete,display,modify,use,list,unlock
```

To allow a user to have all rights on the folder "APPS" and on the folder "APP2" and its sub-folder, but no rights on APP1, specify:

```
folder name=/APPS/ access=@
folder name=/APPS/APP1/APP2 access=@
```

Object type - job

The following table gives the additional access keywords required to work with jobs, other than those described in [Table 41: Access keywords for composer actions on page 221](#):

Table 48. Jobs - additional access keywords

	Activity	Access keywords required
Composer	Use jobs in job streams.	use
Dynamic Workload Console	Also, if a job is used as a recovery job in a job definition, the user must have "use" access to the definition of the job identified as the recovery job.	
Conman	adddep	Add dependencies to jobs in the production plan. Not valid for workstations in end-to-end environment.
Dynamic Workload Console	altpri	Alter the priority of jobs in the production plan. Not valid for workstations in end-to-end environment.
	cancel job	Cancel jobs in the production plan. Not valid for workstations in end-to-end environment.
	confirm	Confirm completion of jobs in the production plan. Not valid for workstations in end-to-end environment.
	deldep job	Delete dependencies from jobs in the production plan. Not valid for workstations in end-to-end environment.
	display	Display jobs in the plan.
	Hold	Hold a job to prevent it from running
	kill	Kill running jobs.

Table 48. Jobs - additional access keywords (continued)

		Activity	Access keywords required
	release job	Release jobs from dependencies in the production plan. Not valid for workstations in end-to-end environment.	release
	reply	Reply to job prompts in the production plan.	reply
	rerun	Rerun jobs in the production plan. Not valid for workstations in end-to-end environment.	rerun; submitdb
		To use the from argument, you must have submitdb access to the job.	
	showjobs	Display information about jobs in the production plan.	list
Conman	submit	Submit commands as jobs or recovery jobs into the production plan.	submit
Dynamic Workload Console	docommand	If the submit also identifies a second job with the "ALIAS" or "RECOVERYJOB" arguments, the user must have "submit" access to that other job, as well Not valid for workstations in end-to-end environment.	
	submit file	Submit files as jobs or recovery jobs into the production plan.	submit
		If the submit also identifies a second job with the "ALIAS" or "RECOVERYJOB" arguments, the user must have "submit" access to that other job as well. Not valid for workstations in an end-to-end environment.	
	submit job	Submit jobs or recovery jobs into the production plan.	submit
		If the submit also identifies a second job with the "ALIAS" or "RECOVERYJOB" arguments, the user must have "submit" access to that other job as well. If the job is defined in a folder, then <code>use</code> access is required on the workstation (<code>cpu</code>) where the job is defined, in addition to access to the folder itself and the objects it contains. Not valid for workstations in an end-to-end environment.	

Table 48. Jobs - additional access keywords (continued)

		Activity	Access keywords required
		<p>Restricts the submission action to jobs defined in the database. With this authorization level a user cannot submit ad hoc jobs. Use this keyword to allow a user to submit only jobs defined in the database. Use the submit keyword to allow a user to submit both defined and ad hoc jobs.</p> <p>Users granted only submitdb rights:</p> <ul style="list-style-type: none"> • Cannot run submit docommand and submit file successfully • Are displayed tasks related to ad hoc job submission on the graphical user interfaces, but if they run them, are returned error messages for lacking the submit access right. 	submitdb
	submit sched	Submit job streams into the production plan. Not valid for workstations in end-to-end environment.	submit
	Hold	Hold a job to prevent it from running	adddep
Dynamic Workload Console	For critical jobs on which you run any of the following actions:	The predecessors are listed regardless of the fact that this authorization might not be extended to them. However, if you want to run any further action on any of the listed predecessors, this will require that you have the proper authorization.	list
	<ul style="list-style-type: none"> • Display hot list • Display critical path • Display incomple ted predecessors • Display completed predecessors 		

Table 48. Jobs - additional access keywords (continued)

		Activity	Access keywords required
Using the workload service assurance feature	All activities	For any user to perform any workload service assurance activities, the <TWS_user> must have the following access keywords:	display, modify, list

Example 1

To allow a user to manage only job dependencies for jobs defined in the root (/) folder, specify:

```
job      access=adddep,deldep
```

Example 2

To allow a user to only manage critical jobs defined in the root (/) folder, specify:

```
job      access=list,altpri
```

Example 3

User `administrator` is granted add, modify, and display rights for all job definitions defined in the folder named "APPS" and any sub-folders, on workstations defined in the root (/) folder, and is therefore permitted to create and modify job definitions that run scripts or commands as needed, with no restriction:

```
USER TWSADMIN
CPU=@+LOGON=administrator
BEGIN
JOB CPU=@ + FOLDER = /APPS + CPUFOLDER = / ACCESS=ADD,MODIFY,DISPLAY,...
[...]
```

User `sconnor` is granted the same rights for jobs that match the condition `jcltype=scriptname`, which means that he can create or modify only job definitions that run scripts and cannot change any of them into a job that runs a command. He can also access all workstation defined in the root (/) folder:

```
USER RESTRICTED
CPU=@+LOGON=sconnor
BEGIN
JOB CPU=@+JCLTYPE=SCRIPTNAME + FOLDER = /APPS + CPUFOLDER = / ACCESS=ADD,MODIFY,DISPLAY,...
[...]
```

Example 4

User `administrator` is granted submit permission for all jobs defined in all folders ("/"), and is therefore permitted to submit jobs defined in the database and ad hoc, with no restriction:

```
USER TWSADMIN
CPU=@+LOGON=administrator
```

```
BEGIN
JOB CPU=@ + FOLDER = / + CPUFOLDER = / ACCESS=ADD,ADDDEP,...,RERUN,SUBMIT,USE,LIST,UNLOCK
[...]
END
```

User `jsmith` is granted `submitdb` permission for all jobs defined in all folders, allowing her to submit all jobs defined in the database, but she is not permitted to run ad hoc job submissions. She also has access to workstations in the `/MYCPUS` folder:

```
USER RESTRICTED
CPU=@+LOGON=jsmith
BEGIN
JOB CPU=@ + FOLDER = / + CPUFOLDER = /MYCPUS ACCESS=ADD,ADDDEP,...,RERUN,SUBMITDB,USE,LIST,UNLOCK
[...]
END
```

Object type - parameter

The following table gives the access keywords required to work with parameters:



Note: Starting from version 8.5, the `parameter` keyword is reserved for parameters created and managed in a local parameter database with the `parms` utility command. See the *IBM Workload Scheduler: User's Guide and Reference* for details on `parms`.

Table 49. Parameters - additional access keywords

	Activity	Access keywords required
<code>parms</code>	Manage local parameter definitions.	<code>display</code>

Example

To allow a user to perform all activities on parameters and with access to all workstations defined in the root (`/`) folder, specify:

```
parameter + folder = / + cpufolder = / access=@
```

Object type - prompt

The following table gives the additional access keywords required to work with prompts, other than those described in [Table 41: Access keywords for composer actions on page 221](#):

Table 50. Prompts - additional access keywords

		Activity	Access keywords required
Composer		Use prompts when defining or submitting jobs and job streams	use
Dynamic Workload Console			
Conman	adddep	Use prompts when adding dependencies to jobs in the production plan. Not valid for workstations in end-to-end environment.	use
Dynamic Workload Console	recall	Display prompts waiting for a response.	display
	reply	Reply to a job or Job Scheduler prompt.	reply
	showprompts	Display information about prompts.	list
	submit docommand	Use prompts when submitting commands as jobs into the production plan. Not valid for workstations in end-to-end environment.	use
	submit file	Use prompts when submitting files as jobs into the production plan. Not valid for workstations in end-to-end environment.	use
	submit job	Use prompts when submitting jobs into the production plan. Not valid for workstations in end-to-end environment.	use
	submit sched	Use prompts when submitting job streams into the production plan. Not valid for workstations in end-to-end environment.	use

Example

To allow a user to perform all activities on prompts except reply to them, specify:

```
prompt      access=use,display,list
```

Object type - report

The following table gives the access keywords required to work with reports.

Table 51. Files- access keywords

		Activity	Access keywords required
Dynamic Workload Console		Display reports on page 212 on Dynamic Workload Console.	display

Example

To allow a user to display reports on the Dynamic Workload Console, specify:

```
report          access=display
```

Object type - resource

The following table gives the additional access keywords required to work with resources, other than those described in [Table 41: Access keywords for composer actions on page 221](#):

Table 52. Resources - additional access keywords

		Activity	Access keywords required
Composer		Use resources when defining or submitting jobs and job streams	use
Dynamic Workload Console			
Conman	adddep	Use resources when adding dependencies to jobs in the production plan. Not valid for workstations in end-to-end environment.	use
Dynamic Workload Console			
	resource	Change the number of units of a resource on a workstation.	resource
	showresources	Display information about resources.	list
	submit	Use resources when submitting commands as jobs into the production plan. Not valid for workstations in end-to-end environment.	use
	docommand	Use resources when submitting commands as jobs into the production plan. Not valid for workstations in end-to-end environment.	use
	submit file	Use resources when submitting files as jobs into the production plan. Not valid for workstations in end-to-end environment.	use
	submit job	Use resources when submitting jobs into the production plan. Not valid for workstations in end-to-end environment.	use
	submit sched	Use resources when submitting job streams into the production plan. Not valid for workstations in end-to-end environment.	use

Example

To allow a user to display information about resources defined in the root folder, and change the units of a resource on a workstation defined in the root folder (/), but not to use them in any other scheduling objects or actions, specify:

```
resource + folder = / + cpufolder = / access=list,resource
```

Object type - run cycle group

The following table gives the access keywords required to work with run cycle groups:

Table 53. Run cycle groups- access keywords

Activity		Access keywords required
Composer	Use run cycle groups in job streams.	use
Dynamic Workload		
Console		

Example

To allow a user to create and delete a run cycle group, specify:

```
runcygrp      access=add,delete
```

Object type - schedule

The following table gives the additional access keywords required to work with job streams, other than those described in [Table 41: Access keywords for composer actions on page 221](#):

Table 54. Job streams - additional access keywords

Activity		Access keywords required
Conman	adddep	Add dependencies to job streams in the production plan. Not valid for workstations in end-to-end environment.
Dynamic Workload		
Console	altpri	Alter the priority of job streams in the production plan. Not valid for workstations in end-to-end environment.
	cancel sched	Cancel job streams in the production plan. Not valid for workstations in end-to-end environment.
	deldep sched	Delete dependencies from job streams in the production plan. Not valid for workstations in end-to-end environment.
	display	Display job streams in the plan. .
	limit sched	Modify the limit for jobs concurrently running within a Job Scheduler.
	release sched	Release job streams from dependencies in the production plan. Not valid for workstations in an end-to-end environment.

Table 54. Job streams - additional access keywords (continued)

		Activity	Access keywords required
	reply	Reply to job stream prompts in the production plan.	reply
	showschedules	Display information about job streams in the production plan.	list
	submit sched	Submit job streams into the production plan.	submit
		If the submit also identifies a second job stream with the "ALIAS" argument, the user must have "submit" access to that other job stream as well.	
		If the job stream is defined in a folder, then <code>use</code> access is required on the workstation (<code>cpu</code>) where the job stream is defined, in addition to access to the folder itself and the objects it contains.	
		Not valid for workstations in an end-to-end environment.	
Using the workload service assurance feature	All activities	For any user to perform any workload service assurance activities, the <code><TWS_user></code> must have the following access keywords:	display, modify, list

Example

To allow a user to perform all actions on job streams defined in the "test" folder and its sub-folders, except submit and release, and access to all workstations defined in the root (/) folder, specify:

```
schedule folder = /test + CPUFOLDER = / access=adddep,altpri,cancel,deldep,display,
limit,reply,list
```

Object type - userobj

The following table gives the additional access keywords required to work with users, other than those described in [Table 41: Access keywords for composer actions on page 221](#):

Table 55. Users - additional access keywords

		Activity	Access keywords required
Composer Dynamic Workload Console	Modeling of job types with advanced options	When defining job types with advanced options allows the modeler to specify in the credentials section of the job that the <code>user name</code> and <code>password</code> values required to submit the job are resolved at run time with values extracted from the database and defined with the	use

Table 55. Users - additional access keywords (continued)

		Activity	Access keywords required
		User definition composer commands (<code>username</code> and <code>password</code>) or Dynamic Workload Console panel.	
		Note that on dynamic agents User definitions can be used regardless of the operating system.	
Conman	<code>altpass</code>	Alter user passwords in the plan.	<code>altpass</code>
Dynamic Workload Console			

Example

The following access definition allows a user to:

- List and modify user information, including passwords in the database (`display`, `modify`, and `altpass`).
- When defining job types with advanced options on dynamic agents, to specify in the credentials section of the job that the `user name` and `password` values required to submit the job are resolved at run time with values extracted from the database and defined with the User definition (`use`).

```
userobj      access=display,modify,altpass,use,list
```

Object type - vartable

The following table gives the access keywords for using variable tables and the variables they contain (this includes the global variables)

Table 56. Variable tables - access keywords

		Activity	Access keywords required
Composer		Use variable tables in run cycles, run cycle groups, job streams, and workstations	<code>use</code>
Dynamic Workload Console			

Example

To allow a user only to use variable tables when defining other scheduling objects, specify:


```
vartable      access=use
```

Object type - workload application

The following table gives the access keywords required to work with workload applications:

Table 57. Workload applications - access keywords

		Activity	Access keywords required
Dynamic Workload Console	add	Add new workload applications templates to the database. Unlock access is needed to use the <code>unlock</code> attribute.	add, unlock
	create	Create a workload application template in the database. Modify access is needed to use the <code>lock</code> attribute.	display, modify
	delete	Delete a workload application template from the database.	delete
	display	Display a workload application template.	display
	list	List workload application templates in the database.	list
	lock	Lock workload application templates in the database.	modify
	modify	Modify a workload application template in the database.	add, modify
	new	Create a workload application template in the database.	add, modify
	rename	Rename workload application templates in the database. The user needs add access to the new object and delete and display access to the old object.	add, delete, display
	replace	Replace workload application templates in the database. Unlock access is needed to use the <code>unlock</code> attribute.	add, modify, unlock
unlock	Unlock workload application templates in the database.	unlock	

Example

To allow a user to create and delete a workload application, specify:

```
wkldappl      access=add,delete
```

The <TWS_user> - special security file considerations

The <TWS_user> is a special user, and requires special consideration for the security file.

Required access for the <TWS_user> for workload service assurance

For any user to perform Workload service Assurance activities, the <TWS_user> must have *display*, *modify* and *list* access keywords assigned for all *job*, *schedule* and *cpu* objects.

New <TWS_user> in migrated Security file

If you change the <TWS_user> of your environment, for example, as you might do when performing a parallel upgrade, and then you migrate the Security file (to preserve your settings) you must set up the new <TWS_user> in the Security file in advance, with all its required access rights, before attempting to start IBM Workload Scheduler.

Update definitions for Windows domain <TWS_user> in the Security file after upgrade to version 9.5

Due to new support of the UPN Windows user, if you have Windows domain users that are defined in the logon fields as `domain\username`, after performing an upgrade to version 9.5, update the Security file before starting the IBM Workload Scheduler instance. Insert the escape character `\` before the `\` character in the `domain\username` value.

For example, if you use the `MYDOMAIN\user1` value in the logon field, after the upgrade, in the Security file you must update the line in following way:

```
.....
logon=MYDOMAIN\user1
.....
```

Sample security file

This section contains a sample security file divided into sections for each different class of user.

Note that the order of definitions is from most to least-specific. Because of the order, *TWS_users* and **root** users are matched first, followed by users in the **sys** group, and then users in the **mis** group. All other users are matched with the last definition, which is the least specific.

TWS_users and root users logged in on the master domain manager

user mastersm cpu=\$master + logon=<TWS_user>,root

```
#####
#   Sample Security File
#####
# APPLIES TO TWS_users AND ROOT USERS LOGGED IN ON THE
# MASTER DOMAIN MANAGER.
user mastersm  cpu=$master + logon=<TWS_user>,root
begin
# OBJECT      ATTRIBUTES          ACCESS CAPABILITIES
# -----
job           cpu=@ + folder = / + cpufolder = /  access=@
schedule     cpu=@ + folder = / + cpufolder = /  access=@
resource     + folder = / + cpufolder = /      access=@
prompt       + folder = /                access=@
file         access=@
calendar     + folder = /                access=@
```

```

cpu          cpu=@ + folder = /          access=@
parameter   name=@ ~ name=r@ + folder = / + cpufolder = / access=@
userobj     cpu=@ + logon=@ + cpufolder = / access=@
eventrule   name=@ + folder = / access=add,delete,display,modify,list,unlock
action      provider=@ access=display,submit,use,list
event       provider=@ access=use
report      name=@ access=display
runcygrp    name=@ + folder = / access=add,delete,display,modify,use,list,unlock
varlable    name=a@,$default + folder = / access=add,delete,display,modify,use,list,unlock
wkldapll    name=@ + folder = / access=add,delete,display,modify,list,unlock
lob         name=@ access=use
folder     name=/ access=@
end

```

This user definition applies to GUI and CLI access for *TWS_users* and **root** users logged into a master domain manager. They are given unrestricted access to all objects, except parameters that have names beginning with **r**. Access to the **r** parameters is given only to users in the **mis** group. They are the only ones who can generate all kinds of plans and who can create, update, and delete event rule definitions.

All users have access to all variable tables beginning with "a" and to the default table, irrespective of the default variable table name.

TWS_users and root users logged in on any domain manager (other than the master)

user testerlondon cpu=\$manager + logon=<TWS_user>,root

```

#####
# Sample Security File
#####
# APPLIES TO TWS_users AND ROOT USERS LOGGED IN ON ANY
# DOMAIN MANAGER.
user testerlondon cpu=$manager + logon=<TWS_user>,root
begin
# OBJECT      ATTRIBUTES      ACCESS CAPABILITIES
# -----
job           cpu=@ + folder = / + cpufolder = / access=add,delete,display
schedule     cpu=@ + folder = / + cpufolder = / access=add,delete,display
resource     + folder = / + cpufolder = / access=@
prompt       + folder = / access=@
file         name=prodsked access=build, display
file         name=trialsked access=build, display
calendar     + folder = / access=@
cpu          cpu=@ + folder = / access=@
parameter   name=@ ~ name=v@ + folder = / + cpufolder = / access=@
userobj     cpu=@ + logon=@ + cpufolder = / access=@
eventrule   name=@ + folder = / access=add,delete,display,modify,list,unlock
action      provider=@ access=display,submit,use,list
event       provider=@ access=use
report      name=@ access=display
runcygrp    name=@ + folder = / access=add,delete,display,modify,use,list,unlock
varlable    name=a@,$default + folder = / access=add,delete,display,modify,use,list,unlock
wkldapll    name=@ + folder = / access=add,delete,display,modify,list,unlock
lob         name=@ access=use
folder     name=/ access=@
end

```

This user definition applies to GUI and CLI access for *TWS_users* and **root** users logged into any domain manager other than the master. They are given unrestricted access to all objects, except parameters that have names beginning with **v**, and jobs and jobs streams to which they have limited access. They can access all workstations defined in the root folder (/). They can generate all types of plans and can create, update, and delete event rule definitions defined in the root folder.

All users have access to all variable tables beginning with "a" and to the default table, irrespective of the default variable table name.

TWS_users and root users logged in on any workstation other than any domain manager

user sm ~CPU=\$MANAGER logon=<TWS_user>,root

```
#####
# APPLIES TO TWS_users AND ROOT USERS LOGGED IN ON ANY
# WORKSTATION OTHER THAN THE MASTER DOMAIN MANAGER.
user sm logon=<TWS_user>,root
begin
# OBJECT      ATTRIBUTES      ACCESS CAPABILITIES
# -----
job           cpu=$thiscpu + folder = / + cpufolder = / access=@
schedule     cpu=$thiscpu + folder = / + cpufolder = / access=@
resource     cpu=$thiscpu + folder = / + cpufolder = / access=@
prompt       + folder = /      access=@
calendar     + folder = /      access=@
cpu          cpu=$thiscpu + folder = / access=@
parameter    cpu=$thiscpu ~ name=r@ + folder = / + cpufolder = / access=@
action       provider=@        access=display,submit,use,list
event        provider=@        access=use
report       name=RUNHIST,RUNSTATS access=display
runcygrp     name=@ + folder = / access=add,delete,display,modify,use,list,unlock
file         name=globalopts  access=display
lob          name=@           access=use
folder       name=/myfolder   access=@
end
```

This user definition applies to *TWS_users* and **root** users to whom definition (1) does not apply, which are those who are logged in on any workstation other than the master domain manager or any other domain manager. They are given unrestricted access to all objects on their login workstation. Note that prompts, files, and calendars are global in nature and are not associated with a workstation.

They can use event rules, but are not allowed to create, update, or delete event rule definitions.

Users logged into the sys group on the master domain manager

user masterop cpu=\$master + group=sys

```
#####
# APPLIES TO USERS LOGGED INTO THE SYS GROUP ON THE
# MASTER DOMAIN MANAGER.
user masterop cpu=$master + group=sys
begin
# OBJECT      ATTRIBUTES      ACCESS CAPABILITIES
# -----
job           cpu=@ + logon="TWS_domain\<TWS_user>"
```

```

+ folder = / access=@
job      cpu=@ + logon=root + folder = /
        + cpufolder = /
        access=adddep,altpri,cancel,confirm,
        deldep,release,reply,rerun,submit,use
job      cpu=@ + logon=@ ~ logon=root + folder = /
        + cpufolder = /
        access=add,adddep,altpri,cancel,confirm,deldep,
        release,reply,rerun,submit,use
schedule cpu=$thiscpu + folder = / + cpufolder = / access=@
schedule cpu=@ + folder = / + cpufolder = /
        access=adddep,altpri,cancel,
        deldep,limit,release,submit
resource + folder = / access=add,display,resource,use
file     name=globalopts access=display
file     name=prodsked access=display
file     name=symphony access=display
file     name=trialsked access=build, display
calendar + folder = / access=display,use
cpu      cpu=@ + folder = / access=@
parameter name=@ ~ name=r@ + folder = / access=@
report   name=RUNHIST,RUNSTATS access=display
wkldappl name=@ + folder = / access=add,delete,display,modify,list,unlock
lob      name=@ access=use
folder   name=/ access=@
end

```

This user definition applies to users logged into the **sys** group on the master domain manager. They are given a unique set of access capabilities. Multiple object statements are used to give these users specific types of access to different sets of objects. For example, there are three job statements:

- The first job statement permits unrestricted access to jobs that run on any workstation (@) under the user's name (*TWS_domain\<TWS_user>*).
- The second job statement permits specific types of access to jobs that run on any workstation and that run as **root**.
- The third job statement permits specific types of access to jobs that run on any workstation. Jobs that run as root are excluded.

They are the only users defined on the master domain manager, different from maestro or root, who can generate trial and forecast plans.

Users logged into the sys group on any workstation other than the master domain manager

user op ~cpu=\$master group=sys

```

#####
# APPLIES TO USERS LOGGED INTO THE SYS GROUP ON ANY
# WORKSTATION OTHER THAN THE MASTER DOMAIN MANAGER
user op group=sys
begin
# OBJECT      ATTRIBUTES      ACCESS CAPABILITIES
# -----
job          cpu=$thiscpu + logon=@ + folder = /

```

```

                                + cpufolder = / access=@
job      cpu=$thiscpu + logon=root + folder = /
                                + cpufolder = /
                                access=adddep,altpri,cancel,confirm,deldep,
                                release,reply,rerun,submit,use
job      cpu=$thiscpu ~ logon=root + folder = /
                                + cpufolder = /
                                access=adddep,altpri,cancel,confirm,deldep,
                                release,reply,rerun,submit,use
schedule cpu=$thiscpu + folder = / + cpufolder = / access=@
resource + folder = /          access=add,display,resource,use
runcygrp name=@ + folder = /   access=add,delete,display,modify,use,list,unlock
prompt   + folder = /          access=add,display,reply,use
calendar + folder = /          access=use
cpu      cpu=$thiscpu + folder = / access=console,fence,limit,
                                link,start,stop,unlink
parameter name=@ ~ name=r@ + folder = / access=@
wkldappl name=@ + folder = /   access=add,delete,display,modify,list,unlock
lob      name=@          access=use
folder   name=/          access=@
end

#####

```

This user definition applies to **sys** group users to whom definition (3) does not apply, which are those who are logged in on any workstation other than the master domain manager. They are given a set of access capabilities similar to those in definition (3). The exception is that access is restricted to objects on the user's login workstation (**\$thiscpu**).

Users logged into the *mis* group on any workstation

user misusers group=mis

```

#####
# APPLIES TO USERS LOGGED INTO THE MIS GROUP ON
# ANY WORKSTATION.
user misusers cpu=@          group=mis
begin
# OBJECT      ATTRIBUTES      ACCESS CAPABILITIES
# -----
job          cpu=$thiscpu + folder = /
            + logon=@ + cpufolder = /      access=@
job          cpu=$thiscpu + folder = /
            + logon=@
            ~ logon=root + cpufolder = /      access=submit,use
schedule     cpu=$thiscpu + folder = /
            + cpufolder = / access=add,submit,modify,display
cpu          cpu=@ + type=agent,s-agent,fta + folder = /
            access=console,fence,limit,link,start,stop,unlink
parameter    name=r@ + folder = / + cpufolder = / access=@
parameter    name=@ + folder = / + cpufolder = / access=display
runcygrp     name=@ + folder = / access=add,delete,display,modify,use,list,unlock
folder       name=/          access=@
end

#####

```

This user definition applies to users logged into the **mis** group on workstations defined in the root folder. They are given a limited set of access capabilities to fault-tolerant, standard, and dynamic agents. Resources, prompts, files, calendars, and workstations are omitted, which prevents access to these objects. These users are given unrestricted access to parameters with names that begin with **r**, and that are defined in the root folder, but can only display other parameters.

Users logged into multiple groups [continue keyword]

This is an example of a security file where the `continue` keyword is used. This kind of security file allows a user to inherit authorization from multiple *stanzas*. The user gets the accesses for the first matching entry of each *stanza* that matches the user definition.

user misusers cpu@ group=mis

```
#####
# User misusers USER DEFINITION APPLIES TO USERS LOGGED IN TO
# THE MIS GROUP ON ANY WORKSTATION.
#
# User dbusers USER DEFINITION APPLIES TO USERS LOGGED IN TO
# THE DB GROUP ON ANY WORKSTATION.
#
# User default USER DEFINITION APPLIES TO ALL USERS.
#

user misusers cpu=@          group=mis
begin
# OBJECT      ATTRIBUTES      ACCESS CAPABILITIES
# -----
job           cpu=@ + name=mis@   + folder = /
              + cpufolder = / access=@
schedule     name=mis@ + folder = / + cpufolder = / access=@
parameter    name=mis@ + folder = / + cpufolder = / access=@
continue
folder       name=/          access=@

user dbusers  cpu=@          group=db
begin
# OBJECT      ATTRIBUTES      ACCESS CAPABILITIES
# -----
job           cpu=@ + name=db_@ + folder = /
              + cpufolder = / access=@
schedule     name=db_@ + folder = /
              + cpufolder = / access=@
parameter    name=db_@ + folder = / + cpufolder = / access=@
continue
folder       name=/          access=@

user default  cpu=@ + logon=@
begin
# OBJECT      ATTRIBUTES      ACCESS CAPABILITIES
# -----
parameter    name=@ + folder = / + cpufolder = / access=display
folder       name=/          access=@
end
```

```
#####
```

Users that belong only to the *mis* group get access to all objects that have a name starting with the *mis* prefix, as specified in the `user misusers` user definition. In addition, the `user default` user definition gives them display access to all parameters.

Users that belong only to the *db* group get access to all objects that have a name starting with the *db_* prefix, as specified in the `user dbusers` user definition. In addition, the `user default` user definition gives them display access to all parameters.

Users that belong to both the *mis* and the *db* groups get access to the objects that have a name starting with the *mis* prefix and to the objects that have a name starting with the *db_* prefix, as specified in the `user misusers` and in the `user dbusers` user definitions. In addition, the `user default` user definition gives them display access to all parameters. Access to jobs, job streams, workstations, and parameters is limited to only those defined in the root (/) folder.

You must order definitions from most specific to least specific. The `user default` user definition gives generic accesses, and must be therefore specified at the end of the file.

All other users logged in on any workstation

`user default cpu=@ + logon=@`

```
#####
# APPLIES TO ALL OTHER USERS LOGGED IN ON ANY
# WORKSTATION.
user default  cpu=@ + logon=@
begin
# OBJECT      ATTRIBUTES          ACCESS CAPABILITIES
# -----
job           cpu=@ + folder = / + cpufolder = / access=@
schedule     cpu=@ + folder = / + cpufolder = / access=@
resource     + folder = / + cpufolder = /          access=@
prompt       + folder = /          access=@
file         + folder = /          access=@
calendar     + folder = /          access=@
cpu          cpu=@ + folder = / access=@
parameter    name=@ ~ name=r@ + folder = / + cpufolder = / access=@
userobj      cpu=@ + logon=@ + cpufolder = / access=@
eventrule    name=@ + folder = / access=add,delete,display,modify,list,unlock
action       provider=@      access=display,submit,use,list
event        provider=@      access=use
report       name=@          access=display
runcygrp     name=@          access=add,delete,display,modify,use,list,unlock
vartable     name=a@,$default + folder = / access=add,delete,display,modify,use,list,unlock
wkldaplr    name=@ + folder = / access=add,delete,display,modify,list,unlock
lob          name=@          access=use
folder       name=/          access=@
end
#####
```

They are given unrestricted access to all objects, except parameters that have names beginning with *r*. They are the only ones who can generate all kinds of plans and who can create, update, and delete event rule definitions. All users have access to all variable tables beginning with "a" and to the default table, irrespective of the default variable table name. Access to most scheduling objects is limited to those defined in the root (/) folder.

All domain1.com windows users logged in on any workstation

user cpu=@ + logon =@\@domain1.com

```
#####
# APPLIES TO ALL OTHER USERS IN THE 'domain1.com' INTERNET DOMAIN LOGGED IN ON ANY
# WORKSTATION.
user default  cpu=@ + logon=@\@domain1.com
begin
# OBJECT      ATTRIBUTES          ACCESS CAPABILITIES
# -----
job           cpu=@ + logon =a@\@domain1.com + folder = /
              + cpufolder = /   access=display
job           cpu=@      + folder = /
              + cpufolder = /   access=@
schedule     + folder = / + cpufolder = /   access=@
resource     + folder = / + cpufolder = /   access=@
prompt       + folder = /                          access=@
file         + folder = /                          access=@
calendar     + folder = /                          access=@
cpu          cpu=@      + folder = /   access=@
parameter    name=@ ~ name=r@ + folder = / + cpufolder = /   access=@
userobj      cpu=@ + logon=@ + cpufolder = /   access=@
eventrule    name=@      + folder = /   access=add,delete,display,modify,list,unlock
action       provider=@      access=display,submit,use,list
event        provider=@      access=use
report       name=@          access=display
runcygrp     name=@      + folder = /   access=add,delete,display,modify,use,list,unlock
varlable     name=g@,$default + folder = /   access=add,delete,display,modify,use,list,unlock
wkldappl     name=@      + folder = /   access=add,delete,display,modify,list,unlock
lob          name=@          access=use
folder       name=/          access=@
end
#####
```

Windows Users in domain1.com whose name begins with 'a' can display only jobs and can manage parameters which name does not begin with r. All other domain1.com Windows users that are logged in on any workstation are given access to all objects defined in the root (/) folder, and to parameters that have names beginning with r. They are the only ones who can generate all kinds of plans and who can create, update, and delete event rule definitions. All users have access to all variable tables beginning with "g" and to the default table, irrespective of the default variable table name.

All MYWINDOM windows users logged in on any workstation

user default cpu=@ + logon=MYWINDOM\@

```
#####
# APPLIES TO ALL "MYWINDOM" WINDOWS USERS LOGGED IN ON ANY
# WORKSTATION.
user default  cpu=@ + logon=MYWINDOM\@
begin
# OBJECT      ATTRIBUTES          ACCESS CAPABILITIES
# -----
job           cpu=@      + folder = /
              + cpufolder = /   access=@
schedule     cpu=@      + folder = /
              + cpufolder = /   access=@
```

```

resource      + folder = / + cpufolder = /   access=@
prompt       + folder = /               access=@
file         + folder = /               access=@
calendar     + folder = /               access=@
cpu          cpu=@ + folder = /       access=@
parameter    name=@ + folder = / + cpufolder = / access=@
userobj      cpu=@ + logon =MYWINDOM\@ + cpufolder = / access=display
userobj      cpu=@ + logon=@ + cpufolder = / access=@
eventrule    name=@ + folder = /     access=add,delete,display,modify,list,unlock
action       provider=@             access=display,submit,use,list
event        provider=@             access=use
report       name=@                 access=display
runcygrp     name=@ + folder = /       access=add,delete,display,modify,use,list,unlock
variable     name=g@,$default + folder = / access=add,delete,display,modify,use,list,unlock
wkldappl     name=@ + folder = /       access=add,delete,display,modify,list,unlock
lob          name=@                 access=use
folder       name=/                 access=@
end
#####

```

Windows Users in `MYWINDOM` whose name begins with 'r' can display only userjobs. All others `MYWINDOM` Windows user that are logged in on any workstation are given unrestricted access to all objects. Access to workstations is limited to workstations defined in the root (/) folder. Access to scheduling objects that can be defined in folders is limited to the root (/) folder, as specified. For example, access to prompts is limited to prompts defined in the root folder `prompt + folder = / access=@`. They are the only ones who can generate all kinds of plans and who can create, update, and delete event rule definitions. All users have access to all variable tables beginning with "g" and to the default table, irrespective of the default variable table name.



Note: Starting with version 9.2, due to support of the Windows users in User Principal Name (UPN) format, you have to specify the windows domain users in a different way in the Security file. In the same example for the previous version you have the following syntax:

```

user default  cpu=@ + logon=MYWINDOM\@
.....
userjob      cpu=@ + logon =MYWINDOM\@ access=display

```

Security file on the master domain manager to install fix packs or upgrade fault-tolerant agents and dynamic agents

```
user MAESTRO CPU=@+LOGON=tws94user,Administrator
```

```

#####
# APPLIES TO tws94user and Administrator LOGGED IN ON ANY WORKSTATIONS.
#####
USER MAESTRO
  CPU=@+LOGON=tws94user,Administrator
BEGIN
  USEROBJ CPU=@ + cpufolder = / ACCESS=ADD,DELETE,DISPLAY,MODIFY,USE,ALTPASS,LIST,UNLOCK
  JOB     CPU=@ + folder = / + cpufolder = /
          ACCESS=ADD,ADDDEP,ALTPRI,CANCEL,CONFIRM,DELDEP,DELETE,DISPLAY,
          KILL,MODIFY,RELEASE,REPLY,RERUN,SUBMIT,USE,LIST,UNLOCK,SUBMITDB,RUN
  SCHEDULE CPU=@ + folder = / + cpufolder = /
          ACCESS=ADD,ADDDEP,ALTPRI,CANCEL,DELDEP,DELETE,DISPLAY,LIMIT,

```

```

                MODIFY,RELEASE,REPLY,SUBMIT,LIST,UNLOCK
RESOURCE CPU=@ + folder = / + cpufolder = / ACCESS=ADD,DELETE,DISPLAY,MODIFY,
                RESOURCE,USE,LIST,UNLOCK
PROMPT         ACCESS=ADD,DELETE,DISPLAY,MODIFY,REPLY,USE,LIST,UNLOCK
FILE  NAME=@   ACCESS=BUILD,DELETE,DISPLAY,MODIFY,UNLOCK
CPU    CPU=@   + folder = /
                ACCESS=ADD,CONSOLE,DELETE,DISPLAY,FENCE,LIMIT,LINK,
                MODIFY,SHUTDOWN,START,STOP,UNLINK,LIST,UNLOCK,RUN,RESETFTA,MANAGE
PARAMETER CPU=@ + cpufolder = / ACCESS=ADD,DELETE,DISPLAY,MODIFY,LIST,UNLOCK
CALENDAR   ACCESS=ADD,DELETE,DISPLAY,MODIFY,USE,LIST,UNLOCK
REPORT     NAME=@ ACCESS=DISPLAY
EVENTRULE  NAME=@ ACCESS=ADD,DELETE,DISPLAY,MODIFY,LIST,UNLOCK
ACTION     PROVIDER=@ ACCESS=DISPLAY,SUBMIT,USE,LIST
EVENT      PROVIDER=@ ACCESS=USE
VARIABLE   NAME=@ ACCESS=ADD,DELETE,DISPLAY,MODIFY,USE,LIST,UNLOCK
WKLDAPPL   NAME=@ ACCESS=ADD,DELETE,DISPLAY,MODIFY,LIST,UNLOCK
RUNCYGRP   NAME=@ ACCESS=ADD,DELETE,DISPLAY,MODIFY,USE,LIST,UNLOCK
LOB        NAME=@ ACCESS=USE
folder     NAME=/ ACCESS=@
END
#####

```

The default MAESTRO definition applies to Dynamic Workload Console and CLI access for `tw94user` and `Administrator` users logged into any workstation in the network. They can install a fix pack or upgrade to a later version fault-tolerant agents and dynamic agents in the network simultaneously.

For more information about this feature, see the section about centralized agent update in *Planning and Installation Guide*.

Chapter 4. Configuring authentication

This section describes how to configure authentication using, amongst other methods, the popular LDAP (Lightweight Directory Access Protocol). It is divided into these main topics:

- [Where to configure authentication on page 252](#)
- [Available configurations on page 252](#)
- [Rules for using a Federated User Registry with IBM Workload Scheduler on page 253](#)
- [Completing the LDAP configuration on page 253](#)
- [Configuring an IBM Tivoli Directory Server on page 255](#)
- [Configuring Microsoft Active Directory on page 258](#)
- [Configuring an OpenID Connect Client on page 262](#)

Where to configure authentication

Authentication must be configured for each WebSphere Application Server Liberty Base profile, following these rules:

To authenticate command-line users

For users of the command-line, the command-line client, and the command-line as clients connected to the master domain manager using HTTP or HTTPS, the same authentication method must be configured for the following components:

- Master domain manager
- Backup master domain manager

To authenticate Z connector users

The Z connector is always installed on the same instance as the Dynamic Workload Console. You do not need to separately configure authentication for it.

To authenticate dynamic domain manager users

The same authentication method must be configured for each dynamic domain manager and its corresponding backup dynamic domain manager. This authentication method does not need to be the same as that used for the master domain manager.

Available configurations

On installation, all IBM Workload Scheduler components that use WebSphere Application Server Liberty Base are configured by default to use a local file-based user repository. For information about supported authentication mechanisms in WebSphere Application Server Liberty Base see the section about authenticating users in WebSphere Application Server Liberty Base.

You can implement an LDAP-based user repository by configuring the sample authentication templates provided in XML format.

If you choose to enable an LDAP-based user repository, for your convenience, a set of sample configuration templates are provided in XML format. See [Configuring IBM Workload Scheduler using templates on page 428](#) for a list of the templates. You can further customize the templates by adding additional elements to the XML files. For a full list of the elements that you can configure to complement or modify the configuration, see the section about LDAP user registry in WebSphere Application Server Liberty Base documentation.

Rules for using a Federated User Registry with IBM Workload Scheduler

This section describes the simple rules you must follow when configuring IBM Workload Scheduler to use a Federated User Registry:

No duplicate User IDs

You can define any number of user registries in a Federated User Registry. However, no user ID must be present in more than one registry and no user ID must be present twice in the same registry. Thus, if you configure multiple user registries it is because you have users in different non-inclusive groups that use different user registries and which need to access IBM Workload Scheduler.

Completing the LDAP configuration

About this task

After you have configured the WebSphere Application Server Liberty Base to use a new authentication configuration, whichever configuration method you used, you must also update the security file, and propagate the changes in your environment.

Updating the security file

About this task

If you use the classic security model, you need to update the IBM Workload Scheduler security file to allow users to access IBM Workload Scheduler objects. For more information, see the section about updating the security file in *Administration Guide*. The following example shows an updated security file, where the user `TEST_LDAP` has been added to the `USER MAESTRO` section:

```
USER MAESTRO
CPU=@+LOGON=tw83,Administrator,administrator,TEST_LDAP
BEGIN
USEROBJ CPU=@ ACCESS=ADD,DELETE,DISPLAY,MODIFY,ALTPASS,UNLOCK,LIST
JOB CPU=@ + FOLDER = / ACCESS=ADD,ADDDEP,ALTPRI,CANCEL,CONFIRM,DELDEP,DELETE,DISPLAY,KILL,
MODIFY,RELEASE,REPLY,RERUN,SUBMIT,USE,LIST,UNLOCK
SCHEDULE CPU=@ + FOLDER = / ACCESS=ADD,ADDDEP,ALTPRI,CANCEL,DELDEP,DELETE,
DISPLAY,LIMIT,MODIFY,RELEASE,REPLY,SUBMIT,LIST,UNLOCK
RESOURCE CPU=@ + FOLDER = / + CPUFOLDER = / ACCESS=ADD,DELETE,DISPLAY,MODIFY,RESOURCE,
USE,LIST,UNLOCK
PROMPT + FOLDER = / ACCESS=ADD,DELETE,DISPLAY,MODIFY,REPLY,USE,LIST,UNLOCK
FILE NAME=@ ACCESS=CLEAN,DELETE,DISPLAY,MODIFY,UNLOCK
CPU CPU=@ + FOLDER = / ACCESS=ADD,CONSOLE,DELETE,DISPLAY,FENCE,LIMIT,LINK,MODIFY,
SHUTDOWN,START,STOP,UNLINK,LIST,UNLOCK
PARAMETER CPU=@ + FOLDER = / + CPUFOLDER = / ACCESS=ADD,DELETE,DISPLAY,MODIFY,UNLOCK,LIST
```

```

CALENDAR      + FOLDER = / ACCESS=ADD,DELETE,DISPLAY,MODIFY,USE,UNLOCK,LIST
              FOLDER  NAME=/ ACCESS=ADD,DELETE,DISPLAY,MODIFY,USE,LIST,UNLOCK, ACL
END

```

In this example, the `useDomainQualifiedUserNames` security property is set to `false` therefore the user name has been specified without the domain.

Propagating the changes

About this task

Propagate the changes you have made as follows:

1. If your changes involved changing the primary WebSphere Application Server Liberty Base administrator, then update the `wa_user.xml` file with the credentials. The `wauser_variables.xml` file can be found in the path:

Dynamic Workload Console

```
DWC_DATA_dir/usr/servers/dwcServer/configDropins/overrides
```

master domain manager

```
TWA_DATA_DIR/usr/servers/engineServer/configDropins/overrides
```

Dynamic Workload Console

```
DWC_home\usr\servers\dwcServer\configDropins\overrides
```

master domain manager

```
TWA_home\usr\servers\engineServer\configDropins\overrides
```

- a. Copy the `wauser_variables.xml` file for both the Dynamic Workload Console and the master domain manager to a temporary directory.
 - b. Create a copy of the original `wauser_variables.xml` file for both the Dynamic Workload Console and the master domain manager in another directory for backup purposes.
 - c. Edit the files in the temporary directory with the updated information about the primary WebSphere Application Server Liberty Base administrator.
 - d. Copy the updated `wauser_variables.xml` files to the `overrides` directory on both the Dynamic Workload Console and the master domain manager.
2. Update the `USERNAME` and `PASSWORD` fields in the `useropts` file on every command-line client that points to your workstation.
 3. Update the `USERNAME` and `PASSWORD` fields in the `useropts` file on every fault-tolerant agent in your environment that has an HTTP/HTTPS connection defined in `localopts` that points to your workstation. The HTTP/HTTPS connection is used to submit a predefined job or job stream.
 4. Update the `USERNAME` and `PASSWORD` fields in the engine connection parameters on every connected Dynamic Workload Console.

Example



Note: To change the `useropts` file, change the USERNAME and type the new PASSWORD in plain text between double quotation marks. The password will be encrypted the first time you log in.

Configuring an IBM Tivoli Directory Server

Enable web single sign-on and use IBM Tivoli Directory Server as an identity provider.

About this task

Client applications, for example, the Dynamic Workload Console, can verify the identity of a user by relying on authentication from IBM Tivoli Directory Server. You can configure the WebSphere Application Server Liberty Base server to function as IBM Tivoli Directory Server to take advantage of web single sign-on and to use IBM Tivoli Directory Server as an identity provider.

To simplify the configuration of the WebSphere Application Server Liberty Base server, a sample configuration file in XML format is provided named `auth_IDS_config.xml`.

Update the configuration file with the details about your identity provider.

- a. Copy the template file to a working directory. The template is located in the following path:

UNIX

```
DWC_DATA_dir/usr/servers/dwcServer/configDropins/templates/authentication
```

Windows

```
DWC_home\usr\servers\dwcServer\configDropins\templates\authentication
```

- b. Edit the template file in the working directory with the desired configuration.
- c. Optionally, create a backup copy of the configuration file `authentication_config.xml` 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.
- d. The `overrides` directory is located in the following path:

UNIX

```
DWC_DATA_dir/usr/servers/dwcServer/configDropins/overrides
```

Windows

```
DWC_home\usr\servers\dwcServer\configDropins\overrides
```

- e. Copy the updated template file to the `overrides` directory, renaming it to `authentication_config.xml` to override the original `authentication_config.xml` file.

Alternatively, if you prefer maintaining the original name of the template, ensure you delete `authentication_config.xml` after you have copied the updated template file to the `overrides` directory to avoid conflicts.

- f. Stop and restart WebSphere Application Server Liberty Base using the `stopappserver` and `startappserver` commands located in `TWA_home/appservertools`.

What to do next

For more detailed information about the IBM Tivoli Directory Server parameters and values to configure in the `auth_IDS_config.xml` file, see the related WebSphere Application Server Liberty Base documentation at [Configuring LDAP user registries in Liberty](#).

Example configurations of LDAP servers for IDS

Refer to this template if you are using an IBM Tivoli Directory Server (IDS). This file describes a default configuration. For more advanced and specific configurations, refer to the relevant WebSphere Application Server Liberty Base documentation at [Configuring LDAP user registries in Liberty](#) or to your LDAP administrator.

IBM Directory Server

```
<server description="federated_basicLDAP">

  <variable name="admin.group.name" value="Admins"/>

  <variable name="ldap.base.DN" value=""/>

  <variable name="ldap.port" value=""/>

  <variable name="ldap.host" value=""/>

  <variable name="ldap.adminDN" value=""/>

  <variable name="ldap.password" value=""/>

  <jndiEntry value="${admin.group.name}" jndiName="admin.group.name" />

  <administrator-role>
    <group>${admin.group.name}</group>
  </administrator-role>

  <federatedRepository searchTimeout="20m">
    <primaryRealm name="TWSRealm" allowOpIfRepoDown="true">
      <participatingBaseEntry name="o=BasicRealm"/>
    </primaryRealm>
  </federatedRepository>
</server>
```



```

        <participatingBaseEntry name="{ldap.base.DN}"/>
        <uniqueGroupIdMapping inputProperty="uniqueName" outputProperty="uniqueName"/>
<groupSecurityNameMapping inputProperty="cn" outputProperty="cn"/>
<groupDisplayNameMapping inputProperty="cn" outputProperty="cn"/>
<userDisplayNameMapping inputProperty="principalName" outputProperty="principalName"/>
<userSecurityNameMapping inputProperty="principalName" outputProperty="principalName"/>
<uniqueUserIdMapping inputProperty="uniqueName" outputProperty="uniqueName"/>
    </primaryRealm>
</federatedRepository>

<ldapRegistry
  baseDN="{ldap.base.DN}"
  ldapType="IBM Tivoli Directory Server"
  port="{ldap.port}"
  host="{ldap.host}"
  id="ldap"
  bindDN="{ldap.adminDN}"
  bindPassword="{ldap.password}"
  searchTimeout="20"
  sslEnabled="false"
  sslRef="twaSSLSettings"
  userFilter="(&(uid=%v)(objectclass=ePerson))"
  groupFilter="(&(cn=%v)(|(objectclass=groupOfNames)
    (objectclass=groupOfUniqueNames)(objectclass=groupOfURLs))"
  userIdMap="*:uid"
  groupIdMap="*:cn"
  groupMemberIdMap="mycompany-allGroups:member;
  mycompany-allGroups:uniqueMember;
  groupOfNames:member;
  groupOfUniqueNames:uniqueMember">
  <ldapEntityType name="Group">
    <objectClass>groupOfNames</objectClass>
  </ldapEntityType>
  <ldapEntityType name="PersonAccount">
    <objectClass>inetOrgPerson</objectClass>
  </ldapEntityType>
  <ldapEntityType name="OrgContainer">
    <objectClass>organization</objectClass>
    <objectClass>organizationalUnit</objectClass>
    <objectClass>domain</objectClass>
    <objectClass>container</objectClass>
  </ldapEntityType>

</ldapRegistry>
basicRegistry id="basic" realm="BasicRealm">

  user name="{user.twsuser.id}" password="{user.twsuser.password}"/>

  group name="{admin.group.name}">
    member name="{user.twsuser.id}"/>
  </group>

```

```
</basicRegistry>

</server>
```

Configuring Microsoft Active Directory

Enable web single sign-on and use Microsoft Active Directory as an identity provider.

About this task

Client applications, for example, the Dynamic Workload Console, can verify the identity of a user by relying on authentication from Microsoft Active Directory. You can configure the WebSphere Application Server Liberty Base server to function as Microsoft Active Directory to take advantage of web single sign-on and to use Microsoft Active Directory as an identity provider.

To simplify the configuration of the WebSphere Application Server Liberty Base server, a sample configuration file in XML format is provided named `auth_AD_config.xml`.

Update the configuration file with the details about your identity provider.

- a. Copy the template file to a working directory. The template is located in the following path:

UNIX

```
DWC_DATA_dir/usr/servers/dwcServer/configDropins/templates/authentication
```

Windows

```
DWC_home\usr\servers\dwcServer\configDropins\templates\authentication
```

- b. Edit the template file in the working directory with the desired configuration.
- c. Optionally, create a backup copy of the configuration file `authentication_config.xml` 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.

- d. The `overrides` directory is located in the following path:

UNIX

```
DWC_DATA_dir/usr/servers/dwcServer/configDropins/overrides
```

Windows

```
DWC_home\usr\servers\dwcServer\configDropins\overrides
```

- e. Copy the updated template file to the `overrides` directory, renaming it to `authentication_config.xml` to override the original `authentication_config.xml` file.

Alternatively, if you prefer maintaining the original name of the template, ensure you delete `authentication_config.xml` after you have copied the updated template file to the `overrides` directory to avoid conflicts.

- f. Stop and restart WebSphere Application Server Liberty Base using the `stopappserver` and `startappserver` commands located in `TWA_home/appservertools`.

What to do next

For more detailed information about Microsoft Active Directory parameters and values to configure in the `auth_AD_config.xml` file, see the related WebSphere Application Server Liberty Base documentation at [Configuring LDAP user registries in Liberty](#).

Example configurations of LDAP servers for Microsoft Active Directory

Refer to this template if you are using Microsoft Active Directory. This file describes a default configuration. For more advanced and specific configurations, refer to the relevant WebSphere Application Server Liberty Base documentation at [Configuring LDAP user registries in Liberty](#) or to your LDAP administrator.

```
<server description="federated_basicLDAP">

<!--
This variable specifies the group name containing the primary DWC's Administrator users.
It can be a group defined in file based userRegistry (into <basicRegistry> section) or in your LDAP-based
directory services authentication.
-->
<variable name="admin.group.name" value="Admins"/>

<!--
The value of your Base distinguished name (DN) of the directory service, which indicates the starting point
for LDAP searches in the directory service.
Sample: <variable name="ldap.base.DN" value="o=domain,c=us"/>
-->
<variable name="ldap.base.DN" value="DC=TWS,DC=COM"/>

<!--
The Port number of the LDAP server.
Sample: <variable name="ldap.port" value="389"/>
-->
<variable name="ldap.port" value="389"/>

<!--
The Address of the LDAP server in the form of an IP address or a domain name service (DNS) name.
Sample: <variable name="ldap.host" value="host.domain.com"/>
-->
<variable name="ldap.host" value="<your_host_name>"/>

<!--
The Distinguished name (DN) for the application server, which is used to bind to the directory service.
Specify a user defined in Microsoft Active Directory Server with look-up rights.
Sample: <variable name="ldap.adminDN" value="cn=testuser,o=domain,c=us"/>
-->
<variable name="ldap.adminDN" value="CN=Operators,DC=TWS,DC=COM"/>
```

```

<!--
The Distinguished name (DN) for the application server, which is used to bind to the directory service.
You can use the liberty provided tool <wlp_dir>/bin/securityUtility to know the encrypted value
of your password.
1. run: <wlp_dir>/bin/securityUtility encode mypassword
2. output: {xor}MiYvPiwsKDA0w==
3. fill the value field with the printed output value
Sample: <variable name="ldap.password" value="{xor}MiYvPiwsKDA0w==" />
-->
<variable name="ldap.password" value="" />

<jndiEntry value="\${admin.group.name}" jndiName="admin.group.name" />

<!-- Assign 'admin' to Administrator -->
  <administrator-role>
    <group>\${admin.group.name}</group>
  </administrator-role>

<!--
Details about how to configure LDAP registry and federate it with basic registry, can be found following this
link:

https://www.ibm.com/support/knowledgecenter/en/SSAW57\_liberty/com.ibm.websphere.wlp.nd.multipatform.doc/ae/twlp\_sec\_ldap.html

https://www.ibm.com/support/knowledgecenter/en/SSEQTP\_liberty/com.ibm.websphere.wlp.doc/ae/cwlp\_repository\_federation.html

To troubleshoot any LDAP authentication issues, copy trace.xml in overrides with the following
traceSpecification:
  traceSpecification="com.ibm.ws.security.wim.*=all:com.ibm.websphere.security.wim.*=all"
-->
<federatedRepository searchTimeout="20m">
  <primaryRealm name="TWSRealm" allowOpIfRepoDown="true">
    <participatingBaseEntry name="o=BasicRealm"/>
    <participatingBaseEntry name="\${ldap.base.DN}"/>
    <uniqueGroupIdMapping inputProperty="uniqueName" outputProperty="uniqueName"/>
    <groupSecurityNameMapping inputProperty="cn" outputProperty="cn"/>
    <groupDisplayNameMapping inputProperty="cn" outputProperty="cn"/>
    <userDisplayNameMapping inputProperty="principalName" outputProperty="principalName"/>
    <userSecurityNameMapping inputProperty="principalName" outputProperty="principalName"/>
    <uniqueUserIdMapping inputProperty="uniqueName" outputProperty="uniqueName"/>
  </primaryRealm>
</federatedRepository>

<!--
Note for LDAP directory service configured in SSL:
1. the settings sslEnabled to "true"
2. Import the LDAP certificate in trustStore used by the server,
   (it is defined in configDropins/defaults/ssl_comfig.xml file, the default one is
   resources/security/TWSServerTrustFile.jks).
   For importing the exported LDAP certificate your_ldap.cert run
   $JAVA_HOME/bin/keytool -import -file ./your_ldap.cert -alias ldapCA -keystore
   resources/security/TWSServerTrustFile.jks
-->
<ldapRegistry id="AD"
  host="\${ldap.host}" port="\${ldap.port}" ignoreCase="true"

```

```

baseDN="{ldap.base.DN}"
bindDN="{ldap.adminDN}"
bindPassword="{ldap.password}"
ldapType="Microsoft Active Directory"
sslEnabled="false"
sslRef="twaSSLSettings">
<activatedFilters
  userFilter="(&!(sAMAccountName=%v)(objectcategory=user))"
  groupFilter="(&!(cn=%v)(objectcategory=group))"
  userIdMap="*:sAMAccountName"
  groupIdMap="*:cn"
  groupMemberIdMap="memberOf:member" >
</activatedFilters>
</ldapRegistry>

<basicRegistry id="basic" realm="BasicRealm">
  <!-- DO NOT DELETE -->
  <user name="{user.twsuser.id}" password="{user.twsuser.password}"/>
  <!-- END DO NOT DELETE -->
  <group name="{admin.group.name}">
    <member name="{user.twsuser.id}"/>
  </group>

  <!-- Sample for adding other users or group in file based user registry. -->
  <!--
  <user name="nonadmin" password="nonadmin"/>
  <user name="analyst" password="analyst"/>
  <user name="developer" password="developer"/>
  <user name="configurator" password="configurator"/>
  <user name="operator" password="operator"/>
  <group name="Admins">
    <member name="{user.twsuser.id}"/>
  </group>
  -->
</basicRegistry>

</server>

```

If you have nested groups in your Microsoft Active Directory, ensure you set the `recursiveSearch` property in the `ldapRegistry id="AD"` section to `true`, as follows:

```

.....
<ldapRegistry id="AD"
  host="{ldap.host}" port="{ldap.port}" ignoreCase="true"
  baseDN="{ldap.base.DN}"
  bindDN="{ldap.adminDN}"
  bindPassword="{ldap.password}"
  ldapType="Microsoft Active Directory"
    recursiveSearch="true"
  sslEnabled="false"
  sslRef="twaSSLSettings">
.....
</ldapRegistry>

```

Configuring an OpenID Connect Client

Enable web single sign-on and use the OpenID Connect Provider as an identity provider.

About this task

Client applications, for example, the Dynamic Workload Console, can verify the identity of a user by relying on authentication from an OpenID Connect Provider. You can configure the WebSphere Application Server Liberty Base server to function as an OpenID Connect Client to take advantage of web single sign-on and to use the OpenID Connect Provider as an identity provider.

To simplify the configuration of the WebSphere Application Server Liberty Base server, a sample configuration file in XML format is provided named `openid_connect.xml`.

Update the configuration file with the details about your identity provider.

- a. Copy the template file to a working directory. The template is located in the following path:

UNIX

```
DWC_DATA_dir/usr/servers/dwcServer/configDropins/templates/authentication
```

Windows

```
DWC_home\usr\servers\dwcServer\configDropins\templates\authentication
```

- b. Edit the template file in the working directory with the desired configuration.
- c. Optionally, create a backup copy of the configuration file `authentication_config.xml` 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.

- d. The `overrides` directory is located in the following path:

UNIX

```
DWC_DATA_dir/usr/servers/dwcServer/configDropins/overrides
```

Windows

```
DWC_home\usr\servers\dwcServer\configDropins\overrides
```

- e. Copy the updated template file to the `overrides` directory, renaming it to `authentication_config.xml` to override the original `authentication_config.xml` file.
Alternatively, if you prefer maintaining the original name of the template, ensure you delete `authentication_config.xml` after you have copied the updated template file to the `overrides` directory to avoid conflicts.
- f. Stop and restart WebSphere Application Server Liberty Base using the `stopappserver` and `startappserver` commands located in `TWA_home/appservertools`.

What to do next

For more detailed information about the OpenID parameters and values to configure in the `openid_connect.xml` file, see the related WebSphere Application Server Liberty Base documentation at [Configuring an OpenID Connect Client in Liberty](#).

Chapter 5. Network administration

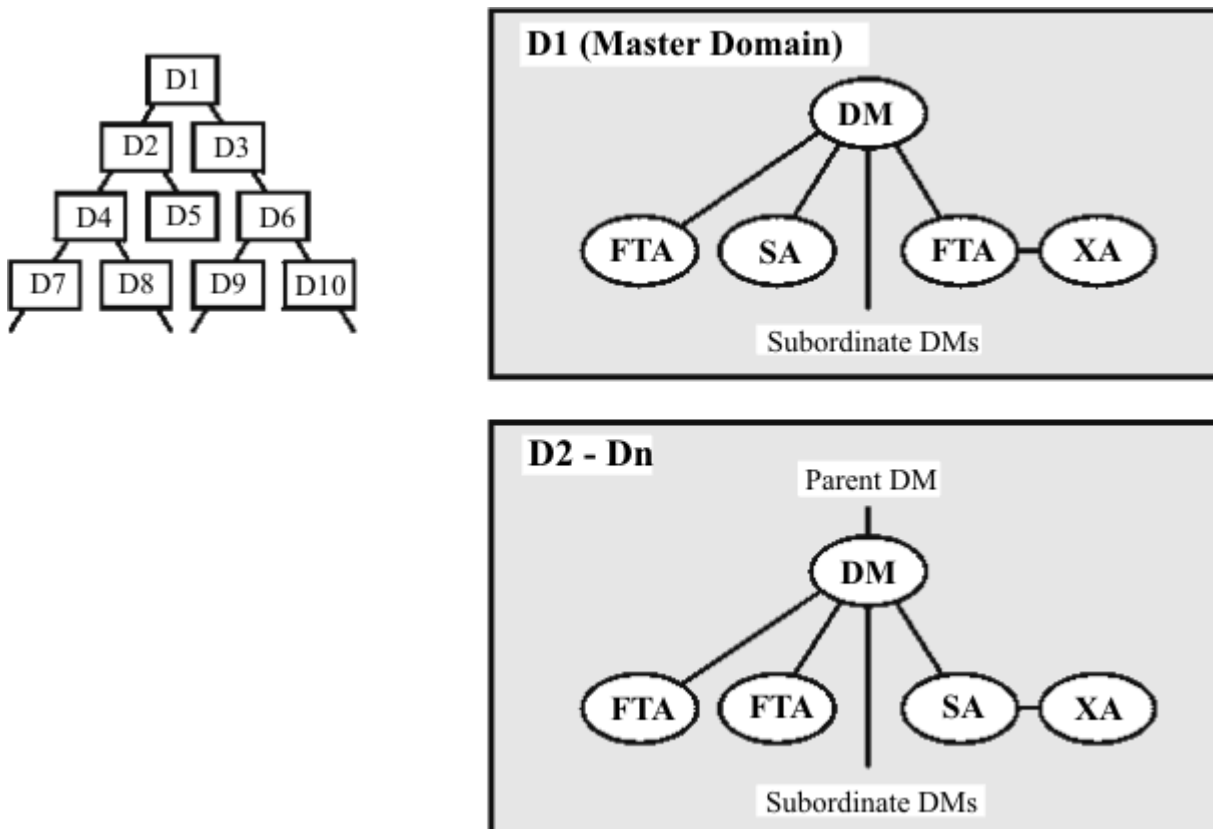
This chapter describes how to administer the IBM Workload Scheduler network. It has the following topics:

- [Network overview on page 264](#)
- [Network definition on page 265](#)
- [Network communications on page 266](#)
- [Network operation on page 273](#)
- [Support for Internet Protocol version 6 on page 296](#)
- [Optimizing the network on page 278](#)
- [Netman configuration file on page 290](#)
- [Defining access methods for agents on page 292](#)
- [IP address validation on page 296](#)
- [Impact of network changes on page 298](#)

Network overview

A IBM Workload Scheduler network consists of one or more domains arranged hierarchically. A IBM Workload Scheduler domain is a logical grouping of workstations, consisting of a domain manager and a number of agents.

Figure 6. IBM Workload Scheduler network domain structure



Network definition

Domain

A named group of IBM Workload Scheduler workstations consisting of one or more agents and a domain manager. All domains have a parent, except the master domain.

Master domain

The topmost domain in an IBM Workload Scheduler network.

Master domain manager

The domain manager in the topmost domain of an IBM Workload Scheduler network. It contains the centralized master files used to document scheduling objects. It creates the Production Control file (Symphony) at the start of each production period and performs all logging and reporting for the network. See also Domain Manager.

Backup master domain manager

A fault-tolerant agent capable of assuming the responsibilities of the master domain manager.

Parent domain

The domain directly above the current domain. All domains, except the master domain, have a parent domain. All communications to/from a domain is rooted through the parent domain manager.

Domain Manager

The management hub in a domain. All communications in and from the agents in a domain is routed through the domain manager. See also Master Domain Manager.

Backup domain manager

A fault-tolerant agent capable of assuming the responsibilities of its domain manager.

Fault-tolerant agent

An agent workstation capable of resolving local dependencies and launching its jobs in the absence of a domain manager.

Standard agent

An agent workstation that launches jobs only under the direction of its domain manager.

Extended agent

An agent workstation that launches jobs only under the direction of its host. Extended agents can be used to interface IBM Workload Scheduler with non-IBM Workload Scheduler systems and applications

Dynamic agent

A workstation that manages a wide variety of job types, for example, specific database or FTP jobs, in addition to existing job types. This workstation is automatically created and registered when you install the dynamic agent. Because the installation and registration processes are performed automatically, when you view the agent in the Dynamic Workload Console, it results as updated by the Resource Advisor Agent. You can group agents in pools and dynamic pools.

In a simple configuration, dynamic agents connect directly to a master domain manager or to a dynamic domain manager. However, in more complex network topologies, if the network configuration prevents the master domain manager or the dynamic domain manager from directly communicating with the dynamic agent, then you can configure your dynamic agents to use a local or remote gateway.

Host

The scheduling function required by extended agents. It can be performed by any IBM Workload Scheduler workstation, except another extended agent.

Network communications

In a IBM Workload Scheduler network, agents communicate with their domain managers, and domain managers communicate with their parent domain managers. There are basically two types of communications that take place:

- Start-of-production period initialization (distribution of new Symphony file)
- Scheduling events in the form of change-of-state messages during the production period

Before the start of each new production period, the master domain manager creates a production control file called *Symphony*. Then, IBM Workload Scheduler is restarted in the network, and the master domain manager sends a copy of the new *Symphony* file to each of its automatically-linked agents and subordinate domain managers. The domain managers, in turn, send copies to their automatically-linked agents and subordinate domain managers. Agents and domain managers that are not set up to link automatically are initialized with a copy of *Symphony* as soon as a link operation is run in IBM Workload Scheduler.

Once the network is started, scheduling messages, like job starts and completions, are passed from the agents to their domain managers, through parent domain managers to the master domain manager. The master domain manager then broadcasts the messages throughout the hierarchical tree to update the *Symphony* files of all domain managers and the domain managers forward the messages to all fault-tolerant agents in their domain running in *FullStatus* mode.

Network links

Links provide communications between IBM Workload Scheduler workstations in a network. Links are controlled by the AUTO Link flag, and the Console Manager **link** and **unlink** commands. When a link is open, messages are passed between two workstations. When a link is closed, the sending workstation stores messages in a local *pobox* file and sends them to the destination workstation when the link is reopened.

This means that when links are closed, the message queues fill up with messages for the inaccessible workstations. To maximize the performance of IBM Workload Scheduler, monitor workstations for closed links and attempt to reopen them as soon as possible.



Note: Extended agents do not have links. They communicate with their domain managers through their hosts.

To have a workstation link opened automatically, turn on the AUTO Link flag in the workstation's definition. The link is first opened when IBM Workload Scheduler is started on the Master Domain workstation. If the subdomain manager

and workstations are not initialized and their AUTO Link flag is on, the master domain manager attempts to link to its subordinates and begin the initialization processes. If the AUTO Link flag is turned off, the workstation is only initialized by running a **link** command from the master domain manager. After the workstation is initialized, it automatically starts and issues a link back to its domain manager.

If you stop a workstation, the links from it to other workstations are closed. However, the links from the other workstations to it remain open until either one of the following situations occurs:

- The stopped workstation is restarted and a **link** command is issued
- The other workstations' **mailman** processes time out, and perform an **unlink** for the workstation

When the **link** command is issued and the connection has been established, if the domain manager does not receive any reply within the timeout period, the `chkhltst` service is automatically invoked by **mailman**.

This service verifies that the workstation mailbox can be successfully read, and checks if there are errors in the mailbox header. Resulting information is logged in the `TWSMERGE.log` file of the domain manager as follows:

- If a file system error occurs while opening the mailbox, the following message is reported: `AWSBDY126E An error occurred opening the Mailbox.msg file in CPU_NAME.`
- If an error occurs while opening the mailbox because **mailman** is reading the mailbox, the following message is reported: `AWSBDY123I The Mailbox.msg file in CPU_NAME is correctly read by Mailman.`
- If the mailbox is correctly opened, but an error occurs while reading the header, the following message is reported: `AWSBDY125E An error occurred reading the header of the Mailbox.msg file in CPU_NAME.`
- If the mailbox is correctly opened and no error occurs while reading the header, the following message is reported: `AWSBDY124W The Mailbox.msg file in CPU_NAME is not read by Mailman.`

This service can also be launched manually by using the **conman** command. See the *IBM Workload Scheduler User's Guide and Reference* for more details.

To be certain that inter-workstation communication is correctly restored, you can issue a **link** command after restarting a workstation.

Working across firewalls

In the design phase of a IBM Workload Scheduler network, the administrator must know where the firewalls are positioned in the network, which fault-tolerant agents and which domain managers belong to a particular firewall, and which are the entry points into the firewalls. When this has been clearly understood, the administrator should define the **behindfirewall** attribute for some of the workstation definitions in the IBM Workload Scheduler database. In particular, if a workstation definition is set with the **behindfirewall** attribute to ON, this means that there is a firewall between that workstation and the IBM Workload Scheduler master domain manager. In this case, the workstation-domain manager link is the only link allowed between the workstation and its domain manager.

All IBM Workload Scheduler workstations should be defined with the **behindfirewall** attribute if the link with the corresponding domain manager, or with any domain manager in the IBM Workload Scheduler hierarchy right up to the master domain manager, is across a firewall.

When mapping an IBM Workload Scheduler network over an existing firewall structure, it does not matter which fault-tolerant agents and which domain managers are on the secure side of the firewall and which ones are on the non secure side. Firewall boundaries should be the only concern. For example, if the master domain manager is in a non secure zone and some of the domain managers are in secured zones, or vice versa, does not make any difference. The firewall structure must always be considered starting from the master domain manager and following the IBM Workload Scheduler hierarchy, marking all the workstations that have a firewall between them and their corresponding domain manager.

For all workstations with **behindfirewall** set to ON, the conman **start** and **stop** commands on the workstation, and the **showjobs** commands are sent following the domain hierarchy, instead of making the master domain manager or the domain manager open a direct connection to the workstation. This makes a significant improvement in security.

This attribute works for multiple nested firewalls as well. For extended agents, you can specify that an extended agent workstation is behind a firewall by setting the **behindfirewall** attribute to ON, on the host workstation. The attribute is read-only in the plan; to change it in the plan, the administrator must update it in the database and then re-create the plan.

See the *IBM Workload Scheduler: User's Guide and Reference* for details on how to set this attribute.

Configuring dynamic agent communications through a gateway

In some complex network topologies, the master domain manager or the dynamic domain manager are prevented from directly communicating with the dynamic agent.

Before you begin

In a simple configuration, dynamic agents connect directly to the master domain manager or to the dynamic domain manager. However, in more complex network topologies, if the network configuration prevents the master domain manager or the dynamic domain manager from directly communicating with the dynamic agent, for example, if the agents are behind a firewall and need to communicate through the internet, or if they need to communicate with a Network Address Translation (NAT) process, then you can configure your dynamic agents to use a local or remote gateway.

About this task

You can set up your dynamic agents to use a gateway for communication with the master domain manager or to the dynamic domain manager when you install a dynamic agent, or you can configure a gateway subsequent to the installation.

For information about the gateway parameters available with the installation of a dynamic agent, see the section about agent installation parameters in *Planning and Installation Guide*.

To configure an existing IBM Workload Scheduler version 9.2 or later dynamic agent to communicate to its master domain manager or dynamic domain manager through a local gateway, perform the following configuration steps:

1. Edit the `JobManager.ini` file on the dynamic agent workstation that you want to configure to communicate through a gateway. Edit the `[ResourceAdvisorAgent]` section so that the value of the **ResourceAdvisorURL** parameter is `https://$(tdwb_server):$(tdwb_port)/ita/JobManagerGW/JobManagerRESTWeb/JobScheduler/resource`, where, `$(tdwb_server)` and `$(tdwb_port)` correspond to the host name and port of the gateway that you want to use for communication with the master domain manager or the dynamic domain manager.
2. Stop and start the dynamic agent to implement the changes.

Results

The master domain manager or dynamic domain manager can now communicate with the dynamic agent workstation through the gateway.

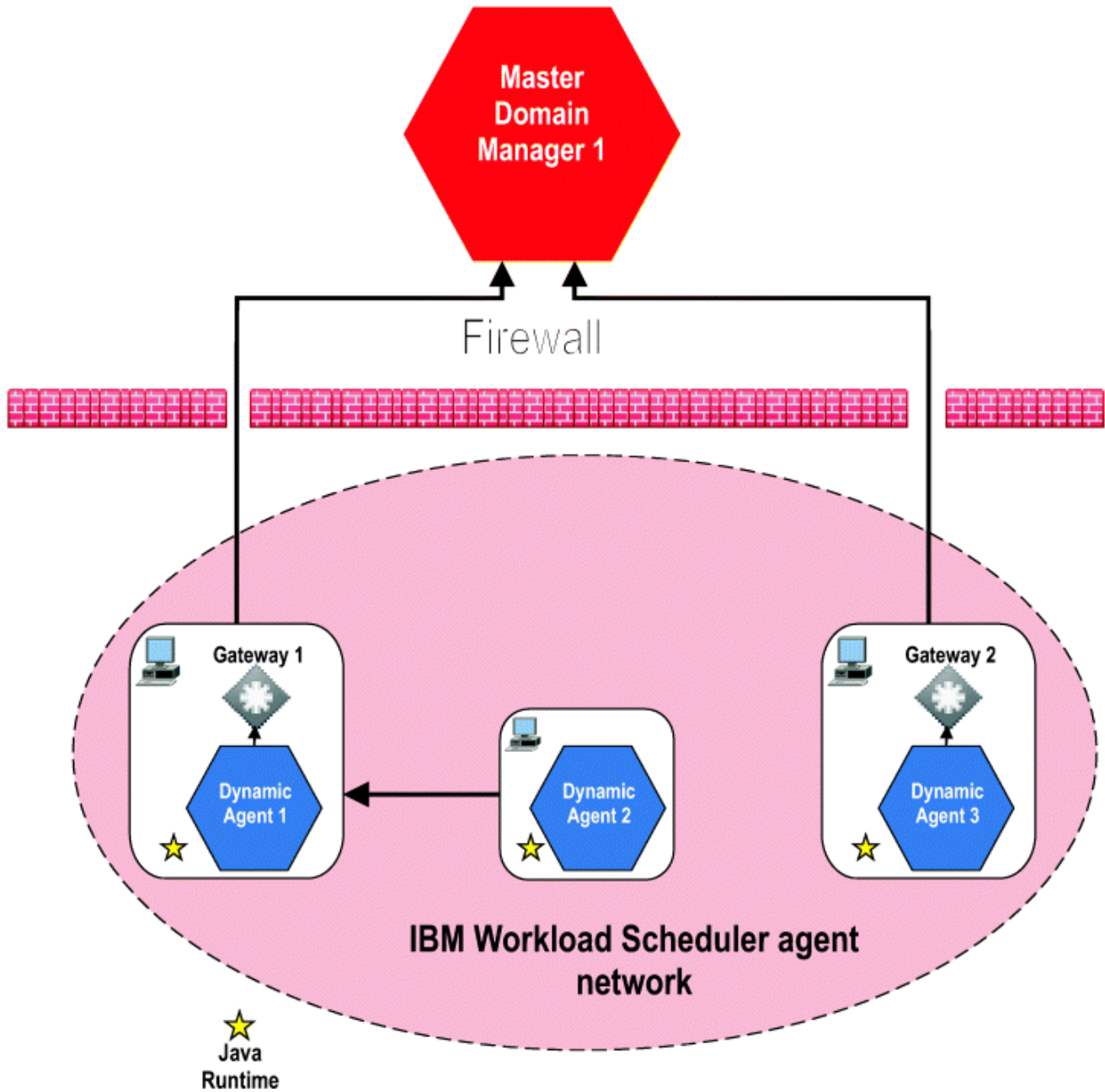


Note:

- If you have more than 100 dynamic agents that communicate through one single gateway to the master domain manager or dynamic domain manager, in the `JobManagerGW.ini` file on the dynamic agent workstation where the gateway resides, set the `ActionPollers` parameter as described in [Configuring general properties \[ITA\] on page 80](#).
- Only for version 9.5 Fix Pack 4, if you install your agents so that they communicate with the master through a remote gateway, ensure that they can reach the master directly at installation time. For more information, see the section about dynamic agent gateway installation examples in *IBM Workload Scheduler: Planning and Installation*

Example

The following diagram depicts a network topology where the master domain manager communicates to the dynamic agents, located behind a firewall, through a gateway configured on one of the dynamic agents.



The following are the configuration settings used in the network topology depicted in the figure:

Table 58. Configuration settings

Dynamic Agent	Configuration File	Parameter	Value
Dynamic Agent 1 - Local gateway	JobManager.ini	Section [ResourceAdvisorAgent] ResourceAdvisorUrl	https:// \$(tdwb_server) : \$(tdwb_port)/ita/ JobManagerGW/

Table 58. Configuration settings (continued)

Dynamic Agent	Configuration File	Parameter	Value
			<pre>JobManagerRESTWeb/ JobScheduler/resource</pre>
			<p>where,</p> <p>\$(tdwb_server)</p> <p>The host name of the Dynamic Agent 1 workstation.</p> <p>\$(tdwb_port)</p> <p>The port number of the Dynamic Agent 1 workstation.</p>
Dynamic Agent 2 - Remote gateway	JobManager.r.ini	<p>Section</p> <p>[ResourceAdvisorAgent]</p> <p>ResourceAdvisorUrl</p>	<pre>https:// \$(tdwb_server): \$(tdwb_port)/ita/ JobManagerGW/ JobManagerRESTWeb/ JobScheduler/resource</pre>
			<p>where,</p> <p>\$(tdwb_server)</p> <p>The host name of the Dynamic Agent 1 workstation.</p> <p>\$(tdwb_port)</p> <p>The port number of the Dynamic Agent 1 workstation.</p>
Dynamic Agent 3 - Local gateway	JobManager.r.ini	<p>Section</p> <p>[ResourceAdvisorAgent]</p> <p>ResourceAdvisorUrl</p>	<pre>https:// \$(tdwb_server): \$(tdwb_port)/ita/ JobManagerGW/ JobManagerRESTWeb/ JobScheduler/resource</pre>
			<p>where,</p> <p>\$(tdwb_server)</p> <p>The host name of the Dynamic Agent 3 workstation.</p> <p>\$(tdwb_port)</p> <p>The port number of the Dynamic Agent 3 workstation.</p>

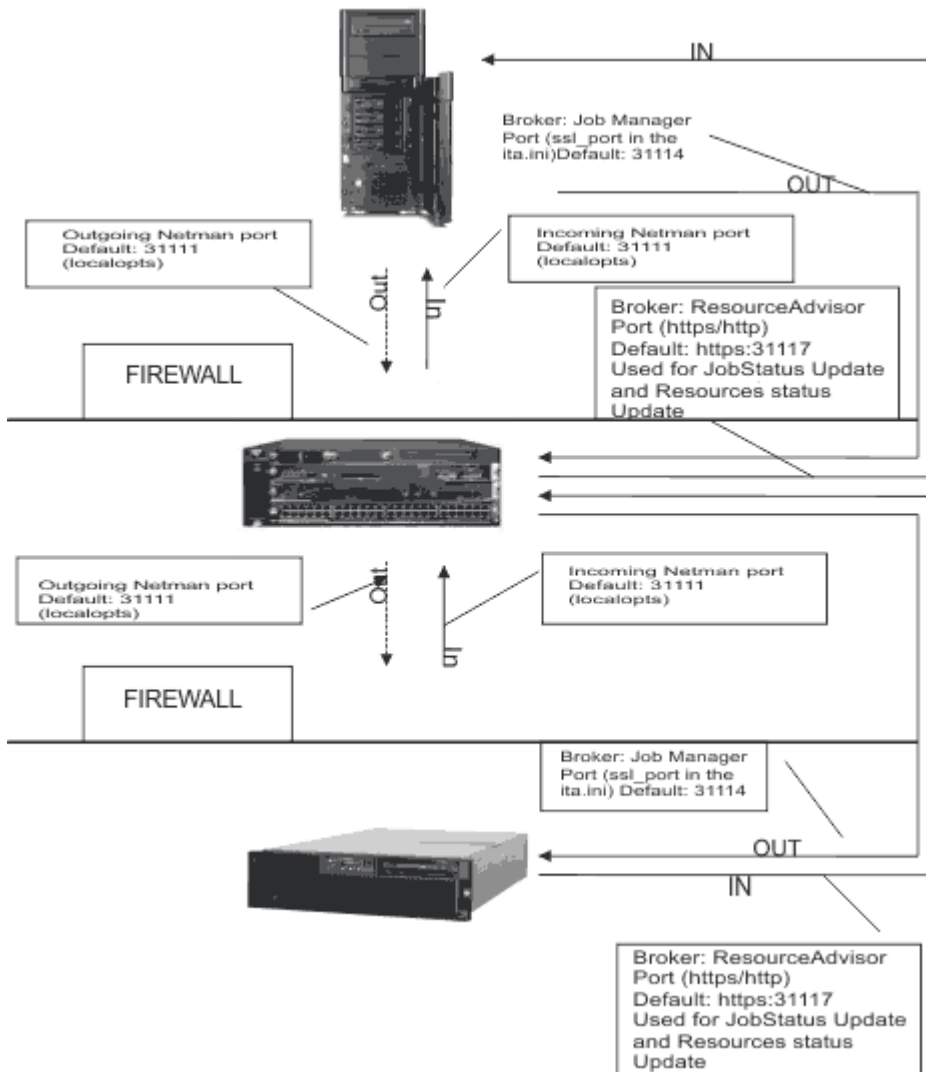
What to do next

For more information about the parameters in the `JobManager.ini` and `JobManagerGW.ini` files, see [Configuring the agent on page 77](#).

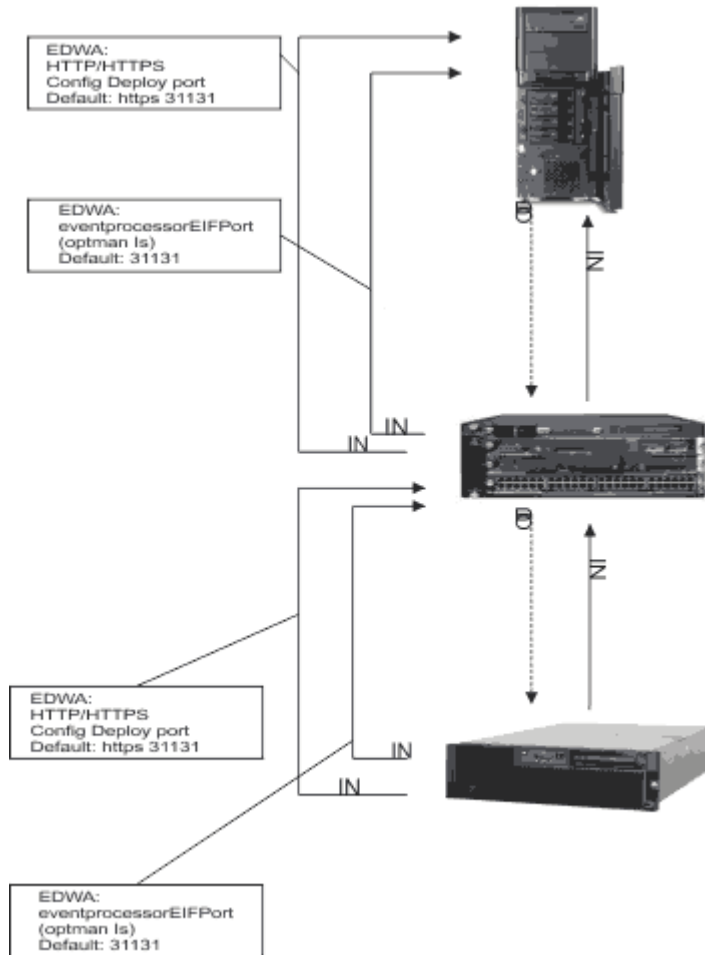
To see an example of the installation parameters that must be specified to configure a gateway when installing a dynamic agent, see the section containing example dynamic agent gateway installations in the *Planning and Installation Guide*.

Enabling Ports

When you install the master domain manager in a IBM Workload Scheduler network all the incoming and outgoing ports are shown in the figure below:



If you enable the event driven workload automation (EDWA) behind the firewall feature the figure below shows all the incoming and outgoing ports.



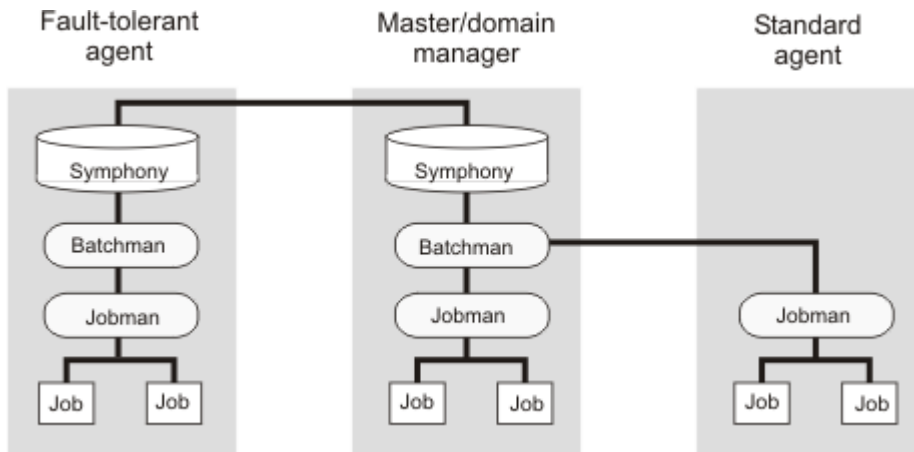
Network operation

The batchman process on each domain manager and fault-tolerant agent workstation operates autonomously, scanning its *Symphony* file to resolve dependencies and launch jobs. Batchman launches jobs via the jobman process. On a standard agent, the jobman process responds to launch requests from the domain manager's batchman.

The master domain manager is continuously informed of job launches and completions and is responsible for broadcasting the information to domain managers and fault-tolerant agents so they can resolve any inter-workstation dependencies.

The degree of synchronization among the *Symphony* files depends on the setting of the *FullStatus* mode in a workstation's definition. Assuming that these modes are turned on, a fault-tolerant agent's *Symphony* file contains the same information as the master domain manager's (see the section that explains how to manage workstations in the database in the *IBM Workload Scheduler: User's Guide and Reference*).

Figure 7. Symphony file synchronization



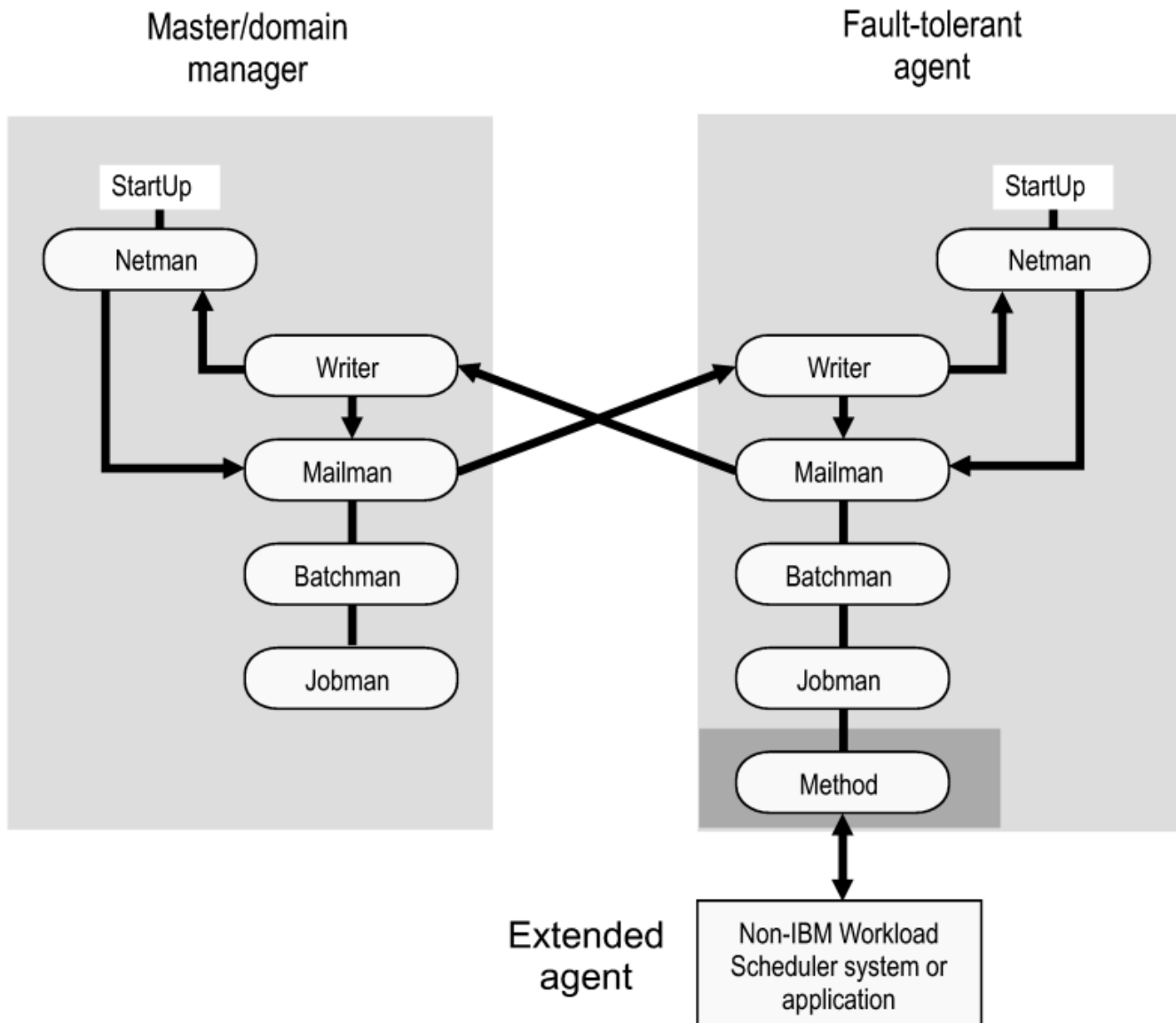
Network processes

Netman is started by the StartUp script (command). The order of process creation is netman, mailman, batchman, and jobman. On standard agent workstations, batchman does not run. All processes, except jobman, run as the **TWS** user. Jobman runs as **root**.

When network activity begins, netman receives requests from remote mailman processes. Upon receiving a request, netman creates a writer process and passes the connection off to it. Writer receives the message and passes it to the local mailman. The writer processes (there might be more than one on a domain manager) are started by link requests and are stopped by unlink requests (or when the communicating mailman terminates).

Domain managers, including the master domain manager, can communicate with a large number of agents and subordinate domain managers. For improved efficiency, you can define mailman servers on a domain manager to distribute the communications load (see the section that explains how to manage workstations in the database in the *IBM Workload Scheduler: User's Guide and Reference*).

Figure 8. Process creation on domain manager and fault-tolerant agent



The StartUp command is normally run automatically, but can also be run manually, as follows:

StartUp

Starts **netman**, the IBM Workload Scheduler network management process.

In Windows™, the **netman** service is started automatically when a computer is restarted. **StartUp** can be used to restart the service if it is stopped for any reason.

In UNIX™, the **StartUp** command can be run automatically by invoking it from the `/etc/inittab` file, so that WebSphere Application Server Liberty Base infrastructure and **netman** is started each time a computer is rebooted. **StartUp** can be used to restart **netman** if it is stopped for any reason.

The remainder of the process tree can be restarted with the

```
conman start
conman startmon
```

commands. See the documentation about conman in the *User's Guide and Reference* for more information.



Note: If you start the StartUp command using a remote shell, the netman process maintains the shell open without returning the prompt. To avoid this problem, modify the StartUp command so that the netman process is called in the background, as follows:

```
# Start netman
/usr/local/TWS95/mae95/TWS/bin/netman&
```

Authorization

You must have **start** access to the workstation.

Syntax

StartUp [-v | -u]

Arguments

-v

Displays the command version and exits.

-u

Displays command usage information and exits.

Example

Examples

To display the command name and version, run the following command:

```
StartUp -v
```

To start the **netman** process, run the following command:

```
StartUp
```

Monitoring the IBM Workload Scheduler processes

You can use event-driven workload automation (EDWA) to monitor the status of network processes and to start a predefined set of actions when one or more specific events take place. For more information about event-driven workload automation, refer to *User's Guide and Reference*.

You can monitor the following processes:

- agent
- appservman
- batchman
- jobman
- mailman
- monman
- netman

The .XML file contains the definition of a sample event rule to monitor the status of the specified processes on the specified workstation. This event rule calls the MessageLogger action provider to write a message in a log file in an internal auditing database. If the condition described in the rule is already existing when you deploy the rule, the related event is not generated. For more information about the MessageLogger action provider, refer to IBM Workload Scheduler User's Guide and Reference:

```
<eventRule name="PROCESSES" ruleType="filter" isDraft="no">
  <eventCondition name="twSProcMonEvt1" eventProvider="TWSApplicationMonitor"
    eventType="TWSProcessMonitor">
    <scope>
      AGENT, BATCHMAN DOWN
    </scope>
    <filteringPredicate>
      <attributeFilter name="ProcessName" operator="eq">
        <value>process_name1</value>
      </attributeFilter>
      <attributeFilter name="TWSPath" operator="eq">
        <value>TWS_path</value>
      </attributeFilter>
      <attributeFilter name="Workstation" operator="eq">
        <value>workstation_name</value>
      </attributeFilter>
      <attributeFilter name="SampleInterval" operator="eq">
        <value>sample_interval</value>
      </attributeFilter>
    </filteringPredicate>
  </eventCondition>
  <action actionProvider="MessageLogger" actionType="MSGLOG" responseType="onDetection">
    <scope>
      OBJECT=AAAAAAA MESSAGE=TWS PROCESS DOWN: %{TWSPROCMONEVT1.PROCESSNAME}
      ON %{TWSPROCMONEVT1.TWSPATH}
    </scope>
    <parameter name="ObjectKey">
      <value>object_key</value>
    </parameter>
    <parameter name="Severity">
      <value>message_severity</value>
    </parameter>
    <parameter name="Message">
      <value>log_message</value>
    </parameter>
  </action>
</eventRule>
</eventRuleSet>
```

where:

process_name

Is the name of the process to be monitored. You can insert more than one process name, as follows:

```
<attributeFilter name="ProcessName" operator="eq">  
  <value>agent</value>  
  <value>batchman</value>  
</attributeFilter>
```

TWS_path

Is the directory containing the Symphony file and the bin directory.

workstation_name

Is the workstation on which the event is generated.

sample_interval

Is the interval, expressed in seconds, for monitoring the process status.

object_key

Is a key identifying the object to which the message pertains.

message_severity

Is the severity of the message.

log_message

Is the message to be logged.

Optimizing the network

The structure of a IBM Workload Scheduler network goes hand in hand with the structure of your enterprise's network. The structure of the domains must reflect the topology of the network in order to best use the available communication channels.

But when planning the IBM Workload Scheduler network, the following must be taken into consideration:

- Data volumes
- Connectivity

Data volumes

Network capacity must be planned to adapt to the amount of data that is circulating. Particularly high transmission volumes might be caused by the following:

- Transfer of large Symphony files.
- Message traffic between the master domain manager and a *FullStatus* agent.
- Message traffic from a domain manager when the domain has many agents.
- Heavy use of internetwork dependencies, which extends traffic to the entire network.

Connectivity

For the more critical agents in your network, you need to consider their position in the network. The reliability of workload execution on a particular agent depends on its capacity to receive a fresh Symphony file at the start of the production period. If the workload contains many dependencies, a reliable connection to the rest of the network is also required. These factors suggest that the best place for critical agents is in the master domain, or to be set up as domain managers immediately under the master domain manager, possibly receiving their Symphony files through a set of dedicated mailman servers. Further, it is important for critical agents that any domain manager above them in the tree structure must be hosted on powerful systems and must have an adequate backup system to ensure continuity of operation in the event of problems.

IBM Workload Scheduler provides two mechanisms to accommodate a particular network situation: the domain structure and mailman servers. Whereas domain structure establishes a hierarchy among IBM Workload Scheduler agents, mailman servers are used to tune the resources dedicated to the connection between two agents.

Domain

Use the IBM Workload Scheduler domain structure mechanism to create a tree-shaped structure for the network, where all communications between two points use the unique path defined by the tree (climb to the common ancestor and go down to the target, as opposed to direct TCP communication). As a consequence, the domain structure separates the network into more-manageable pieces. This is for easier filtering, overview, action, and monitoring. However, it does also introduce some delay in the workload processing. For instance when distributing the Symphony file, a fault-tolerant agent inside a domain needs to wait for two steps of Symphony distribution to be completed (from master domain manager to domain manager and from domain manager to fault-tolerant agent). The same is valid for every other type of communication that comes from the master domain manager.

This has the following implications:

- Critical business activities must be as close as possible to the master domain manager
- The domain manager must be installed on as powerful a workstation as possible
- A similarly powerful backup domain manager must be included in the network
- The network link between the domain manager and its backup must be as fast as possible to pass all the updates received from the subtree
- If intervention is needed directly on the domain, either give shell access to the operators to use the IBM Workload Scheduler command line, or install a connector so that the Dynamic Workload Console can be used.

Mailman servers

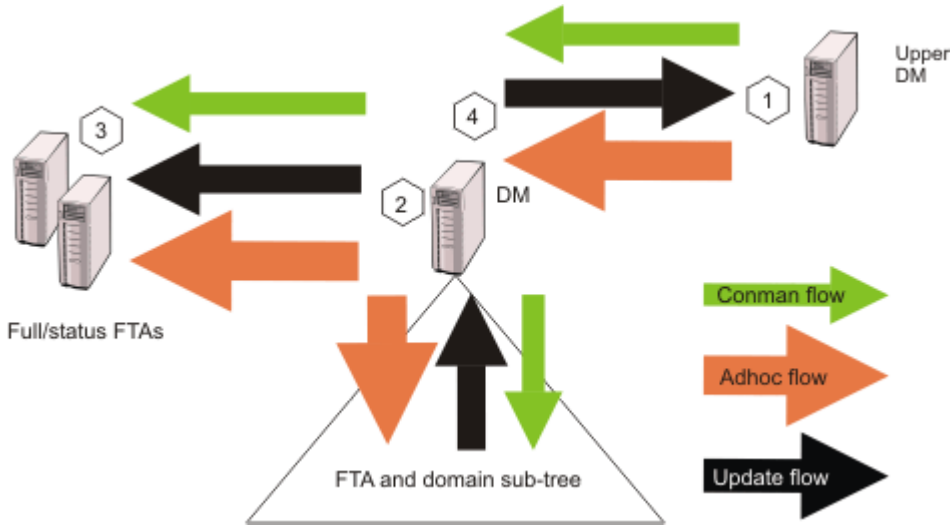
Mailman servers allocate separate processes dedicated to the communication with other workstations. The main mailman is dedicated to the transfer and network hub activities. The use of mailman servers on the domain manager must be carefully planned. The main parameter is the number of downstream connections at each level of the tree. This number describes the number of mailman servers that a main mailman is connected to, or the number of agents a mailman server is connected to. The maximum number of downstream connections is about 20 for Solaris, 50 for Windows™ and about 100 for other UNIX™ workstations, depending

on their power. Typical downstream connections is about 10 for Solaris, about 15 for Windows™ and about 20 for other UNIX™ workstations. However, you must also take into consideration the link speed and the queue sizes, discussed below.

Planning space for queues

In order to plan space for event queues, and possible alert levels and reactions, it is necessary to model the flows passing through the agents, and the domain managers in particular.

Figure 9. Typical IBM Workload Scheduler network flows.



For a typical domain manager, the main flow comes from update activity reported by the sub tree, and from ad hoc submissions arriving from the master domain manager and propagating to the entire network. Under these conditions, the most critical errors are listed by order of importance in [Table 59: Critical flow errors on page 280](#):

Table 59. Critical flow errors

Flow no.	Location	Queue	Risk	Impact
1	Upper domain manager	dm.msg	The queue fills up because of too many unlinked workstations in the domain or a downstream domain manager has failed.	The upper domain manager fails and propagates the error.
2	Domain manager	FullStatus fta.msg	The queue fills because of too many unlinked workstations in the domain or because the FullStatus fault-tolerant agent is not coping with the flow.	The domain manager fails and favors the occurrence of #1.
3	Domain manager and FullStatus fault-tolerant agent	Mailbox.msg or Intercom.msg	The queue fills because the FullStatus fault-tolerant agent cannot cope with flow.	The FullStatus fault-tolerant agent fails and favors the occurrence of #2.

Table 59. Critical flow errors (continued)

Flow no.	Location	Queue	Risk	Impact
4	Domain manager	tomaster.msg	The queue fills because of too many unlinked workstations in the domain.	The domain manager starts to unlink the subtree and accumulates messages in the structure.
5	Fault-tolerant agents - only when <code>enSwfaultTol</code> global option is set to <code>yes</code>	deadletter.msg	The queue fills because of too many unlinked workstations in the domain.	The agent stops.
6	Fault-tolerant agents - only when <code>enSwfaultTol</code> global option is set to <code>yes</code>	ftbox.msg	This queue is circular. The rate of messages entering the queue exceeds the rate of messages being processed, because of too many unlinked workstations in the domain.	Events are lost.

**Note:**

1. Flows are greater at the master domain manager and at any *FullStatus* fault-tolerant agents in the master domain than at subordinate domain managers or *FullStatus* fault-tolerant agents.
2. Use `evtsize -show` to monitor queue sizes.
3. The amount of update flow is related to the amount of workload running in a particular subtree and is unavoidable.
4. The amount of ad hoc flow is related to the amount of additional workload on any point of the network. It can be reduced by planning more workload even if it is inactive. Note that simple reruns (not `rerun from`) do not create an ad hoc flow.

The planning, alert, and recovery strategy must take into account the following points:

- Queue files are created with a fixed size and messages are added and removed in a cyclical fashion. A queue reaches capacity when the flow of incoming messages exceeds the outgoing flow for a sufficient length of time to use up the available space. For example, if messages are being added to a queue at a rate of 1MB per time unit and are being processed and removed at a rate of 0.5 MB per time unit, a queue sized at 10 MB (the default) is at capacity after 20 time units. But if the inward flow rate descends to be the same as the outward flow rate after 19 time units, the queue does not reach capacity.
- The risk of the domain manager failing can be mitigated by switching to the backup domain manager. In this case, the contents of the queues on the domain manager are unavailable until the domain manager backup is started. In

all cases, the size of the queue on the upper domain manager towards any other domain manager must respect the condition A, as indicated in the table [Table 60: Queue sizing conditions. on page 282.](#)

- The risk that fault-switching fault-tolerant agents might not be able to cope with the flow must be planned beforehand. The specifications for fault-switching fault-tolerant agents must be similar to those of the domain manager, to avoid that an agent receives a load that is not appropriate to its capacity. Check if a queue is forming at the *FullStatus* fault-tolerant agents, both in ordinary and in peak operation situations.
- Once risk #2 has been dealt with, the possibility of a network link failure can be mitigated by sizing the queue from a domain manager to the *FullStatus* fault-tolerant agents appropriately as a function of the average network outage duration, and by increasing the size of the mailbox in case of unexpected long outage (see condition B of [Table 60: Queue sizing conditions. on page 282.](#))
- The same condition applies for avoiding an overflow of the domain manager's tomaster.msg queue with respect to network outages (see condition C) of [Table 60: Queue sizing conditions. on page 282.](#)

Table 60. Queue sizing conditions.

A	$\text{MaxAlertTime} \leq \text{size}(\text{UpperDM\#queueToDM}) / \text{averageAdhocFlow}$
B	$\text{MaxNetOutage} \leq \text{size}(\text{DM\#queueToFSFTA}) / (\text{averageAdhocFlow} + \text{averageUpdateFlow})$
C	$\text{MaxNetOutage} \leq \text{size}(\text{DM\#queueToUpperDM}) / \text{averageUpdateFlow}$

Monitoring the IBM Workload Scheduler message queues

You can use event-driven workload automation (EDWA) to monitor the size of message queues and to start a predefined set of actions when one or more specific events take place. For more information about event-driven workload automation, refer to *IBM Workload Scheduler: User's Guide and Reference*.

You can monitor the following message queues:

- appserverbox
- mailbox
- clbox
- intercom
- courier
- monbox
- moncmd
- server
- tomaster
- pobox
- planbox

The following .XML file contains the definition of a sample event rule to monitor the mailbox queue on the specified workstation and send an email when the filling percentage is greater than the specified value. If the condition described in the rule is already existing when you deploy the rule, the related event is not generated. This event rule calls the MailSender

action provider to send an email to the receivers you specify. For more information about the MailSender action provider, refer to *IBM Workload Scheduler: User's Guide and Reference*:

```
<?xml version="1.0"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules"
  xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules
  http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules/EventRules.xsd">
<eventRule name="MONITORQUEUE" ruleType="filter" isDraft="no">
  <eventCondition name="twsMesQueEvt1" eventProvider="TWSApplicationMonitor" eventType="TWSMessageQueues">
    <scope>
      MAILBOX FILLED UP 80% ON FTA
    </scope>
    <filteringPredicate>
      <attributeFilter name="MailboxName" operator="eq">
        <value>mailbox_name</value>
      </attributeFilter>
      <attributeFilter name="FillingPercentage" operator="ge">
        <value>filling_percentage</value>
      </attributeFilter>
      <attributeFilter name="Workstation" operator="eq">
        <value>workstation_name</value>
      </attributeFilter>
      <attributeFilter name="SampleInterval" operator="eq">
        <value>sample_interval</value>
      </attributeFilter>
    </filteringPredicate>
  </eventCondition>
  <action actionProvider="MailSender" actionType="SendMail" responseType="onDetection">
    <scope>
      TWSUSER@TWS : THE MAILBOX ON workstation_name...
    </scope>
    <parameter name="To">
      <value>main_receiver_list</value>
    </parameter>
    <parameter name="Subject">
      <value>mail_subject</value>
    </parameter>
  </action>
</eventRule>
</eventRuleSet>
```

where:

mailbox_name

Is the name of the mailbox to monitor.

filling_percentage

Is the filling percentage. Supported operators are as follows:

ge

causes the event generation when the mailbox filling percentage increases over the threshold value. The event is generated only the first time the specified mailbox filling percentage is reached. If you restart the SSM agent and the filling percentage is higher than the threshold value,

the event is generated again. [Table 61: Example for the ge operator on page 284](#) provides an example in which the **ge** operator is set to 70%.

Table 61. Example for the ge operator

Ma il box name	Filling percentage	A ct ion
Sa m ple (0)	$\geq 70\%$	ev ent not ge ne ra ted
Sa m ple (0)	$< 70\%$	ev ent not ge ne ra ted
Sa m ple (n -1)	$< 70\%$	ev ent not ge ne ra ted
Sa m ple (n)	$\geq 70\%$	ev ent ge ne ra ted
Sa m ple	$\geq 70\%$	ev ent not

Table 61. Example for the ge operator (continued)

Mailbox name	Filling percentage	Action
(n+1)		generated

le

causes the event generation when the mailbox filling percentage decreases under the threshold value. The event is generated only the first time the specified mailbox filling percentage is reached. If you restart the SSM agent and the filling percentage is lower than the threshold value, the event is not generated until the filling percentage increases over the threshold value and then decreases under it again. [Table 62: Example for the le operator on page 285](#) provides an example in which the **le** operator is set to 50%:

Table 62. Example for the le operator

Mailbox name	Filling percentage	Action
Sa m ple (0)	<= 50%	event not generated
Sa m ple (0)	> 50%	event not generated

Table 62. Example for the le operator (continued)

Mailbox name	Filling percentage	Action
Sample (n-1)	Sa > 50%	event not generated
Sample (n)	Sa <= 50%	event generated
Sample (n+1)	Sa <= 50%	event not generated

workstation_name

Is the workstation on which the event is generated.

sample_interval

Is the interval, expressed in seconds, for monitoring the mailbox filling percentage.

main_receiver_list

Is the main receiver list.

mail_subject

Is the subject of the mail.

Changing a queue size

Use the `evtsize` command to resize a queue.

When you have used `evtsize` to resize a queue, the queue remain at that size until the next time you use `evtsize`. It only reverts to the default size of 60 MB if you delete it, at which point IBM Workload Scheduler re-creates it with the default size.

evtsize

Defines the size of the IBM Workload Scheduler message files. This command is used by the IBM Workload Scheduler administrator either to increase the size of a message file after receiving the message, "End of file on events file.", or to monitor the size of the queue of messages contained in the message file.

Authorization

You must be **maestro** or **root** in UNIX™, or **Administrator** in Windows™ to run **evtsize**. Stop the IBM Workload Scheduler engine before running this command.

Syntax

evtsize -V | -U

evtsize *file_name* *size*

evtsize -compact *file_name* [*size*]

evtsize -show *file_name*

Arguments

-V

Displays the command version and exits.

-U

Displays command usage information and exits.

-compact *file_name* [*size*]

Reduces the size of the specified message file to the size occupied by the messages present at the time you run the command. You can optionally use this keyword to also specify a new file size.

-show *file_name*

Displays the size of the queue of messages contained in the message file

file_name

The name of the event file. Specify one of the following:

Administration

```
Courier.msg  
Intercom.msg  
Mailbox.msg  
PlanBox.msg  
Server.msg  
pobox/workstation.msg
```

size

The maximum size of the event file in bytes. When first built by IBM Workload Scheduler, the maximum size is set to 10 MB.



Note: The size of the message file is equal to or bigger than the real size of the queue of messages it contains and it progressively increases until the queue of messages becomes empty; as this occurs the message file is emptied.

Example

Examples

To set the maximum size of the `Intercom.msg` file to 20 MB, run the following command:

```
evtsize Intercom.msg 20000000
```

To set the maximum size of the `pobox` file for workstation `chicago` to 15 MB, run the following command:

```
evtsize pobox\chicago.msg 15000000
```

The following command:

```
evtsize -show Intercom.msg
```

returns the following output:

```
IBM Workload Scheduler (UNIX)/EVTSIZE 9.4 (1.2.2.4) Licensed Materials -  
Licensed Materials - Property of IBM* and HCL**  
5698-WSH  
(C) Copyright IBM Corp. 1998, 2016 All rights reserved.  
(C) Copyright HCL Technologies Ltd. 2016, 2022 All rights reserved.  
* Trademark of International Business Machines  
** Trademark of HCL Technologies Limited  
AWSDEK703I Queue size current 240, maximum 10000000 bytes (read 48, write 288)
```

where:

880

Is the size of the current queue of the `Intercom.msg` file

10000000

Is the maximum size of the `Intercom.msg` file

read 48

Is the pointer position to read records

write 928

Is the pointer position to write records

Tuning mailman servers

Once the distribution of agents to mailman servers has been established, all the groups of agents attached to the same server must respect the link condition.

The link condition relates the number of agents connected to a mailman process and the tuning parameters for unlink on the mailman and writer side.

No_agents(i)

The number of agents connected to a given mailman server *i*

Mm_unlink

A parameter set in the `localopts` of both domain manager and agent. Specifies the maximum number of seconds mailman waits before unlinking from a workstation that is not responding.

Wr_unlink

A parameter set in the `localopts` of both domain manager and agent. Specifies the number of seconds the writer process waits before exiting if no incoming messages are received.

Max_down_agents

The maximum probable number of agents that are unavailable without having the `ignore` flag set in the database and having the `autolink` flag on.

tcp timeout

A parameter set in the `localopts` of both domain manager and agent. Specify the maximum number of seconds that can be waited for the completion of a TCP/IP request on a connected workstation that is not responding.

The condition is:

```
Wr_unlink = Mm_unlink > 1.2 * Max_down_agents * tcp timeout
```

This condition expresses that if the time before unlink is smaller than the probable time of idle waiting of the mailman process (waiting connect timeout for each agent that is currently down) in its loop to reactivate the connections, the agents unlink constantly when some agents are down.

Netman configuration file

The netman configuration file exists on all IBM Workload Scheduler workstations to define the services provided by netman. It is called `<TWA_home>/TWS/network/Netconf`. The `NetConf` file includes comments describing each service. The services are:

2001

Start a writer process to handle incoming messages from a remote mailman.

2002

Start the mailman process. Mailman, in turn, starts the rest of the process tree (batchman, jobman).

2003

Stop the IBM Workload Scheduler process to handle incoming messages from a remote mailman.

2004

Find and return a stdlist file to the requesting Conman process.

2005

Switch the domain manager in a domain.

2006

Locally download scripts scheduled by an IBM® Z Workload Scheduler master domain manager.

2007

Required to bypass a firewall.

2008

Stop IBM Workload Scheduler workstations in a hierarchical fashion

2009

Runs the `switchmgr` script to stop and restart a manager in such a way that it does not open any links to other workstations until it receives the `switchmgr` event. Can only be used when the `enSwfaultTo1` global option is set to `yes`.

2010

Starts mailman with the parameter `demgr`. It is used by the service `2009`. Can only be used when the `enSwfaultTo1` global option is set to `yes`.

2011

Runs `monman` as a child process (`son bin/monman.exe`)

2012

Runs `conman` to stop the event monitoring engine (`command bin/conman.exe stopmon`).

2013

Runs `conman` to switch event processors (`command bin/conman.exe switchevtproc -this`)

2014

Runs conman to start event processing (`command bin/conman.exe startevtproc -this`)

2015

Runs conman to stop event processing (`command bin/conman.exe stopevtproc -this`)

2016

Runs conman to force the update of the monitoring configuration file for the event monitoring engine (`command bin/conman.exe deployconf`)

2017

Runs conman to stop event processing on a client (`client bin/conman.exe synchronizedcmd -stopevtproc`)

2018

Runs conman to check event processing on a client (`client bin/conman.exe synchronizedcmd -checkevtproc`)

2021

Runs conman to start appservman

2022

Runs conman to run the subcommand stopappserver that stops the application server

2023

Runs conman to run the subcommand startappserver that starts the application server

2501

Check the status of a remote job.

2502

Start the Console Manager – a service requested by the client side of the Remote Console. See the *IBM® Tivoli® Remote Control: User's Guide* for more information.

2503

Used by the connector to interact with r3batch extended agent.

Determining internal Symphony table size

The mailman service (2002) can optionally take a parameter that determines the initial size of the internal Symphony table. If you do not supply this parameter, mailman calculates the initial table size based on the number of records in the file.



Note: Mailman expands the table if it needs to, even if this parameter is not supplied.

In normal circumstances, leave mailman to take the default value in the `NetConf` file as supplied (32000). However, if you are experiencing problems with memory, you can allocate a table that is initially smaller. To do this you change the parameter to the service 2002 in the `NetConf` file. The syntax for the entry is:

```
2002    son    bin/mailman [ -parm number ]
```

where, *number* is used to calculate the initial Symphony table size based on the number of records in the Symphony file.

If *r* is the number of records in the Symphony file when batchman starts, [Table 63: Calculation of internal Symphony table on page 292](#) shows how the size of the internal Symphony table is calculated, depending on the value of *number*.

Table 63. Calculation of internal Symphony table

Value of <i>number</i>	Table size
0	$(4/3r) + 512$
n	if $n > r$, n if $n \leq r$, $(4/3r) + 512$
-1	65535
-n	if $+n \Rightarrow r$, n if $+n < r$, $r + 512$

If during the production period you add more jobs, the maximum internal Symphony table size is increased dynamically, up to the maximum number of records allowed in the Symphony file, which is 2,000,000,000.

Defining access methods for agents

Access methods are used to extend the job scheduling functions of IBM Workload Scheduler to other systems and applications. They run on:

Extended agents

They are logical workstations related to an access method hosted by a physical IBM Workload Scheduler workstation (not another extended agent). More than one extended agent workstation can be hosted by the same IBM Workload Scheduler workstation and use the same access method. The extended agent runs on fault-tolerant agents defined using a standard IBM Workload Scheduler workstation definition, which gives the extended agent a name and identifies the access method. The access method is a program that is run by the hosting workstation whenever IBM Workload Scheduler submits a job to an external system.

Jobs are defined for an extended agent in the same manner as for other IBM Workload Scheduler workstations, except that job attributes are dictated by the external system or application.

Information about job running execution is sent to IBM Workload Scheduler from an extended agent using the job `stdlist` file. A method options file can specify alternate logins to launch jobs and check `opens` file dependencies. For more information, see the *User's Guide and Reference*.

A physical workstation can host a maximum of 255 extended agents.

dynamic agents and IBM Z Workload Scheduler agents

They communicate with external systems to start the job and return the status of the job. To run access methods on external applications using dynamic agents, you define a job of type **access method**.

Access methods are available on the following systems and applications.

- SAP
- z/OS
- Custom methods
- unixssh
- unixrsh
- Local UNIX (fault-tolerant agents only)

The UNIX™ access methods included with IBM Workload Scheduler, are described in the related section in *Administration Guide*.

If you are working with dynamic agents, for information about defining IBM Workload Scheduler workstations, see the section that explains how to define workstations in the database in *User's Guide and Reference*. For information about writing access methods, see the section about the access method interface in *User's Guide and Reference*.

More information about access methods is found in *Scheduling Applications with IBM Workload Automation*.

UNIX™ access methods

IBM Workload Scheduler includes two types of UNIX™ access methods, local UNIX access methods and remote UNIX access methods.

The Local UNIX™ access method runs on extended agents. Use the Local UNIX™ access method to enable a single UNIX™ workstation to operate as two IBM Workload Scheduler workstations, both of which you can run IBM Workload Scheduler scheduled jobs.

The Remote UNIX™ access method runs on extended agents and dynamic agents.

On extended agents

Use the Remote UNIX™ access method to designate a remote UNIX™ workstation to run IBM Workload Scheduler scheduled jobs without having IBM Workload Scheduler installed on it.

On dynamic agents

Define a job of type **xajob** that runs on dynamic agents. The dynamic agent communicates with the external system to start the job and return the status of the job.

Local UNIX™ access method running on fault-tolerant agents only

The Local UNIX™ method can be used to define multiple IBM Workload Scheduler workstations on one workstation: the host workstation and one or more extended agents. When IBM Workload Scheduler sends a job to a local UNIX™ extended agent, the access method, **unixlocl**, is invoked by the host to run the job. The method starts by running the standard configuration script on the host workstation (`<TWA_home>/TWS/jobmanrc`). If the logon user of the job is permitted to use a local configuration script and the script exists as `$HOME/TWS/.jobmanrc`, the local configuration script is also run. The job itself is then run either by the standard or the local configuration script. If neither configuration script exists, the method starts the job.

The launching of the configuration scripts, `jobmanrc` and `.jobmanrc` is configurable in the method script. The method runs the configuration scripts by default, if they exist. To disable this feature, you must comment out a set of lines in the method script. For more information, examine the script file `<TWA_home>/TWS/methods/unixlocl` on the extended agent's host.

Remote UNIX™ access method

The Remote UNIX™ access method can be used to designate a non-IBM Workload Scheduler workstation to run jobs scheduled by IBM Workload Scheduler. You can use `unixrsh` or `unixssh`:

The `unixrsh` access method

When IBM Workload Scheduler sends a job to a remote UNIX™ extended agent, the access method, `unixrsh`, creates a `/tmp/maestro` directory on the non-IBM Workload Scheduler workstation. It then transfers a wrapper script to the directory and runs it. The wrapper then runs the scheduled job. The wrapper is created only once, unless it is deleted, moved, or is outdated.

To run jobs using the `unixrsh` access method, the job logon users must be given appropriate access on the non-IBM Workload Scheduler UNIX™ workstation. To give appropriate access, a `.rhost`, `/etc/host.equiv`, or equivalent file must be set up on the workstation. On extended agents, if `opens` file dependencies are to be checked, `root` access must also be permitted. Contact your system administrator for help. For more information about the access method, examine the script file `TWA_home/TWS/methods/unixrsh` on an extended agent's host.

The `unixssh` access method

The `unixssh` access method works like `unixrsh` but uses a secure remote shell to connect to the remote host. The files used by this method are:

```
methods/unixssh
methods/unixssh.wrp
```

The `unixssh` method uses the `ssh` key. You can generate this keyword with any tools that are compatible with the secure remote shell.



Note: The passphrase must be blank.

The following scenario gives an example of how to set up the method:

You installed a IBM Workload Scheduler, fault-tolerant agent or dynamic agent with the *TWS_user*: *twsuser*. You want to run a remote shell in the remote host "REMOTE_HOST" with the user "guest". The procedure is as follows:

1. Create the public and private key for the user *twsuser*, The following is an example using *rsa*:

- a. Log on as *twsuser*

- b. Run

```
ssh-keygen -t rsa
```

- c. When the tool asks for the passphrase, press Enter (leaving the passphrase blank.) The keys are saved as follows:

Key	Location	Comment
Public	<i>TWA_home</i> / <i>TWS</i> / <i>.ssh/id_rsa.pub</i>	
Private	<i>TWA_home</i> / <i>TWS</i> / <i>.ssh/id_rsa</i>	Do not send this file!



Note: Different tools store the key in different places.

2. At the remote host, perform the following actions:

- a. Telnet to the remote host.

- b. Log on as "guest".

- c. Change to the *.ssh* directory in the user home directory, or create it if it does not exist (the directory permissions must be adequate: for example, 700 for the directory and 600 for its contents).

- d. Append the *public* key you created in step 1 to the *authorized_keys* file (create the file if it does not exist), using the command:

```
cat id_rsa.pub >> authorized_keys
```

3. At the fault-tolerant agent or dynamic agent, make the remote host "known" before attempting to let IBM Workload Scheduler processes use the connection. This action can be achieved in one of two ways:

- Log on as *twsuser* and connect to the host using the command:

```
ssh -l guest remote_host_name ls
```

A prompt is displayed saying that the host is not known, and asking permission to access it. Give permission, and the host is added to the list of known hosts.

- Alternatively, use the *ssh* documentation to add the remote host to the file of known hosts.

Managing production for extended agents

In general, jobs that run on extended agents behave like other IBM Workload Scheduler jobs. IBM Workload Scheduler tracks a job's status and records output in the job's `stdlist` files. These files are stored on the extended agent's `host` workstation. For more information on managing jobs, see the section that describes IBM Workload Scheduler plan tasks in the *IBM Workload Scheduler: User's Guide and Reference*.

Failure launching jobs on extended agents and dynamic agents

If the access method is not in the proper directory on the extended agent's host, on the dynamic agent, or the method cannot be accessed by IBM Workload Scheduler, jobs fail to launch or a file dependency is not checked. For a job, the IBM Workload Scheduler jobs logon or the logon specified in the method options file must have read and execute permissions for the access method. When checking a file to satisfy an `opens` dependency, root is used as the login unless another login is specified in the method options file. For more information about method options, see the *IBM Workload Scheduler: User's Guide and Reference*.

IP address validation

When a TCP/IP connection is established, `netman` reads the requester's node name and IP address from the socket. The IP address and node name are used to search the `Symphony` file for a known IBM Workload Scheduler workstation with one of the following possible results:

- If an IP address match is found the validation is considered successful.
- If a node name match is found, the validation is considered successful.
- If no match is found in the `Symphony` file or the IP address returned does not match the one read from the socket, the validation is considered unsuccessful.

The local option, `nm ipvalidate`, determines the action to be taken if IP validation is unsuccessful. If the option is set to `full`, unsuccessful validation causes IBM Workload Scheduler to close the connection and generate an error message. If the option is set to `none` (default), IBM Workload Scheduler permits all connections, but generates a warning message for unsuccessful validation checks.

Support for Internet Protocol version 6

IBM Workload Scheduler supports Internet Protocol version 6 (IPv6) in addition to the legacy IPv4. To assist customers in staging the transition from an IPv4 environment to a complete IPv6 environment, IBM Workload Scheduler provides IP dual-stack support. In other terms, the product is designed to communicate using both IPv4 and IPv6 protocols simultaneously with other applications using IPv4 or IPv6.

To this end, the IPv4-specific `gethostbyname` and `gethostbyaddr` functions have been replaced by the new `getaddrinfo` API that makes the client-server mechanism entirely protocol independent.

The `getaddrinfo` function handles both name-to-address and service-to-port translation, and returns `sockaddr` structures instead of a list of addresses. These `sockaddr` structures can then be used by the socket functions directly. In this way,

`getaddrinfo` hides all the protocol dependencies in the library function, which is where they belong. The application deals only with the socket address structures that are filled in by `getaddrinfo`.

Operating system configuration (UNIX™ only)

IP validation depends on the system call `getaddrinfo()` to look up all the valid addresses for a host. The behavior of this routine varies, depending on the system configuration. When `getaddrinfo()` uses the file `/etc/hosts`, it returns the first matching entry. If the connection is initiated on an address which appears after the first matching entry, IP validation fails. To resolve the problem, place the entry used to initiate the connection before any other matching entries in the `/etc/hosts` file. If `getaddrinfo()` uses the "named" name server or the Network Information Service server and `getaddrinfo()` fails, contact your system administrator for assistance.

IP address validation messages

Following is a list of the messages for IP validation. If the Local Option `nm ipvalidate` is set to `none` (default), the errors appear as warnings.

See the end of the list of conditions for the key to the variables:

- IBM Workload Scheduler workstation name is not found in the Symphony file

```
Ip address validation failed for request:
Service <num> for <program> on <workstation>(<operating_system_type>).
Connection received from IP address:
<c_ipaddr>. MAESTRO CPU <workstation> not found in
Symphony file.
```

- Call to `getaddrinfo()` fails:

```
IP address validation failed for request:
Service num for <program> on cpu(<operating_system_type>).
Connection received from IP address:
<c_ipaddr>. getaddrinfo() failed, unable to
retrieve IP address of connecting node: <node>.
```

- IP Addresses returned by `getaddrinfo()` do not match the IP address of connection workstation:

```
IP address validation failed for request:
Service <num> for <program> on <workstation>(<operating_system_type>).
Connection received from IP address:
<c_ipaddr>. System known IP addresses for node
name node: <k_ipaddr>.
```

- The IP address specified in the workstation definition for the IBM Workload Scheduler workstation indicated in the service request packet does not match the IP address of connecting workstation:

```
IP address validation failed for request:
  Service <num> for <program> on <workstation>(<operating_system_type>).
Connection received from IP address:
<c_ipaddr>. TWS known IP addresses for cpu
<k_ipaddr>.
```

- Regardless of the state of `nm_ipvalidate`, the following information message is displayed when IP validation cannot be performed because the `Symphony` file does not exist or an error occurs when reading it:

```
IP address validation not performed for
request: Service <num> for <program> on
<workstation>(<operating_system_type>). Connection received from IP
address: <c_ipaddr>. Cannot open or read
Symphony file. Service request accepted.
```

Where:

<num>

Service number (2001-**writer**, 2002-**mailman**...)

<program>

Program requesting service

<workstation>

IBM Workload Scheduler workstation name of connecting workstation

<operating_system_type>

Operating system of connecting workstation

<node>

Node name or IP address of connecting workstation

<c_ipaddr>

IP address of connecting workstation

<k_ipaddr>

Known IP address for connecting workstation

IP validation is always successful in the absence of a `Symphony` file. In communications from a domain manager to an agent it is normally successful because a `Symphony` file does not yet exist. However, if the agent has a `Symphony` file from a previous run, the initial link request might fail if the `Symphony` file does not include the name of the domain manager.

Impact of network changes

Any changes that you make to your network might have an impact on IBM Workload Scheduler. Workstations can be identified within IBM Workload Scheduler by host name or IP address. Any changes to host names or IP addresses of

specific workstations must obviously be also implemented in the IBM Workload Scheduler database. However, remember that if those workstations are involved in jobs that are currently scheduled in the Symphony file, those jobs are looking for the old workstation identity.

Changes to host names or IP addresses of specific workstations can be activated immediately by running `JnextPlan -for 0000`. A new production plan is created (containing the updated IP addresses and host names), but the plan time span is not extended.

Thus, plan any network changes with the job schedules in mind, and for major changes you are advised to suspend IBM Workload Scheduler activities until the changes complete in the network and also implemented in the IBM Workload Scheduler database.

Network changes also have a specific impact on the connection parameters used by the application server and the command-line client:

Application server

If you change the network you will need to change the communication parameters specified in the application server configuration files. How to do this is described in the appendix on the utilities supplied with the WebSphere Application Server Liberty Base in the *Planning and Installation Guide*.

Command-line client

When you connect from the command-line client you supply a set of connection parameters. This is done in one of these ways:

From the `localopts` file

The default method is that the connection parameters in the `localopts` file are customized when the command line client is installed.

From the `useropts` file

A `useropts` file might have been created for the user in question, containing a version of the connection parameters personalized for the user.

In the command line, individually

When you invoke one of the command-line programs, you can optionally include the parameters as arguments to the command. These override the values in the `localopts` or `useropts` files.

In the command line, in a file

When you invoke one of the command-line programs, you can optionally include the parameters in a file, the name of which is identified as the `-file` argument to the command. These override the values in the `localopts` or `useropts` files.

Modify whichever method you are using to incorporate the new network connection details.

Chapter 6. Connection security overview

IBM Workload Scheduler provides a secure, authenticated, and encrypted connection mechanism for communication based on the Secure Sockets Layer (SSL) protocol, which is automatically installed with IBM Workload Scheduler.

IBM Workload Scheduler also provides default certificates to manage the SSL protocol that is based on a private and public key methodology.

If you do not customize SSL communication with your certificates, to communicate in SSL mode, IBM Workload Scheduler uses the default certificates that are stored in the default directories, as explained in [SSL connection by using the default certificates on page 302](#). However, in a production environment, it is recommended that you customize SSL communication with your own certificates.

Starting from Version 9.5, Fix Pack 3, you can optionally generate your SSL certificates automatically when you perform a fresh installation from the CLI using either `.jks` or `.PEM` certificates, as described in the sections about Installing the master domain manager and backup master domain manager, Installing the Dynamic Workload Console servers, and Installing agents.

When you perform a fresh installation, you only need to provide either `.jks` or `.PEM` certificates, specify the directory where the files are located and the password you want to use for the keystore and truststore.

Starting from Version 9.5, Fix Pack 4, you can optionally download certificates in `.PEM` format from the master domain manager to your agent.

When installing the agent with a fresh installation, you only need to provide the credentials to connect to the master domain manager using the **wauser** and **wapassword** parameters. The certificates in `.PEM` format are automatically downloaded and deployed to the agent without further intervention.

If you have previously installed the agent, you can run the `AgentCertificateDownloader` script on the agent. The script connects to the master domain manager, downloads the certificates in `.PEM` format, and deploys them to the agent. The certificates must be available on the master domain manager in a specific path. For more information, see the section about the `AgentCertificateDownloader` script in *IBM Workload Scheduler: Planning and Installation*.

The installation program automatically generates the certificates. However, SSL communication between fault-tolerant agents is not enabled by default at installation time, and must be manually configured afterwards. For more information on how to configure SSL for fault-tolerant agents, see [Scenario: SSL Communication across the fault-tolerant agent network on page 317](#).

Consider that using `.jks` and `.kdb` files is supported but not recommended because it involves several manual steps, which might lead to errors, while the automatic procedure with `.PEM` files is the recommended method.



Note: Only for version 9.5 Fix Pack 4, if you install your agents so that they communicate with the master through a remote gateway, ensure that they can reach the master directly at installation time. For more information, see the section about dynamic agent gateway installation examples in *IBM Workload Scheduler: Planning and Installation*.

If you are upgrading from a previous version or did not use the SSL parameters when performing a fresh installation of Version 9.5, Fix Pack 3 or later, you can customize SSL communication with your own certificates as explained in the following scenarios:

- [Customizing certificates for master domain manager and dynamic agent communication on page 303](#)
- See the scenario about connection between the Dynamic Workload Console and the IBM Workload Scheduler components in *Planning and Installation Guide*.
- [Customizing certificates for master domain manager and Dynamic Workload Console communication on page 311](#)
- [Extending communication scenarios to other server components on page 315](#)
- [Scenario: SSL Communication across the fault-tolerant agent network on page 317](#)
- [Command Reference on page 327](#)

Creating a Certificate Authority

How to create a CA and generate the key

About this task

If you do not have a corporate CA, you can perform the sample steps listed below to create one and generate the key, by modifying the data to match your environment. You can use the openssl command located in the installation directory, as follows:

1. Browse to the following path:

On Windows operating systems

```
inst_dir\TWS\bin\tmpopenssl
```

On UNIX operating systems

```
inst_dir/TWS/tmpOpenSSL64/1.1/bin/openssl
```

2. `./openssl genrsa -out ca.key 2048`
3. `./openssl req -x509 -new -nodes -key ca.key -subj "/CN=WA_ROOT_CA" -days 3650 -out ca.crt -config ./openssl.cnf`

The `ca.key` must remain secret, the `ca.crt` is involved in the procedure.

4. `./openssl genrsa -des3 -out tls.key 2048`
5. `./openssl req -new -key tls.key -out tls.csr -config ./openssl.cnf`
6. `./openssl x509 -req -in tls.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out tls.crt`
7. Retrieve the `tls.crt` and `tls.key`.

SSL connection by using the default certificates

The SSL connection between the console and other product components is enabled by using the default certificates.

Before you begin

Dynamic Workload Console and master domain manager, backup master domain manager, dynamic domain manager, backup dynamic domain manager or agent is enabled by using the default certificates.

You can also create certificates starting from your .PEM files, as described in [Connection security overview on page 300](#)

About this task

You have the following environment:

Dynamic Workload Console installed on the *DWC-WKS* workstation:

- The Dynamic Workload Console is installed in the <DWC_INST_DIR> directory.

Master domain manager, backup master domain manager, dynamic domain manager, backup dynamic domain manager, or agent installed on the *TWS-WKS* workstation:

- The agent is installed in the <TWS_INST_DIR> directory.

By default the SSL connection between the Dynamic Workload Console and the component is enabled by using the default certificates. The default password associated with each of the default keystores is `default`. The SSL connection has the following default certificates:

The master domain manager uses two keystores in .jks format: a private key keystore and a trusted key keystore:

On Windows systems

Private keys keystore

```
<TWA_home>\usr\servers\engineServer\resources\security
\TWSServerKeyFile.jks
```

Trusted keys keystore

```
<TWA_home>\usr\servers\engineServer\resources\security
\TWSServerTrustFile.jks
```

On UNIX systems

Private keys keystore

```
<TWA_DATA_DIR>/usr/servers/engineServer/resources/security/
TWSServerKeyFile.jks
```

Trusted keys keystore

```
<TWA_DATA_DIR>/usr/servers/engineServer/resources/security/
TWSServerTrustFile.jks
```

The dynamic agent uses two keystores, one in CMS format (.kdb) and a copy of this one in .jks format. Both keystores contain both the private certificate and the trusted keys:

On Windows systems

.kdb keystore

```
<TWA_home>\TWS\ITA\cpa\ita\cert\TWSCliantKeyStore.kdb
```

.jks keystore

```
<TWA_home>\TWS\ITA\cpa\ita\cert\TWSCliantKeyStoreJKS.jks
```

On UNIX systems

.kdb keystore

```
<TWA_DATA_DIR>/ITA/cpa/ita/cert/TWSCliantKeyStore.kdb
```

.jks keystore

```
<TWA_DATA_DIR>/ITA/cpa/ita/cert/TWSCliantKeyStoreJKS.jks
```



Note: The default certificates are not used for the Dynamic Workload Console client authentication. Authentication on the Client is managed by a user ID and password.

Customizing certificates for master domain manager and dynamic agent communication

Supported scenarios for creating custom certificates for communication between master domain manager and dynamic agent



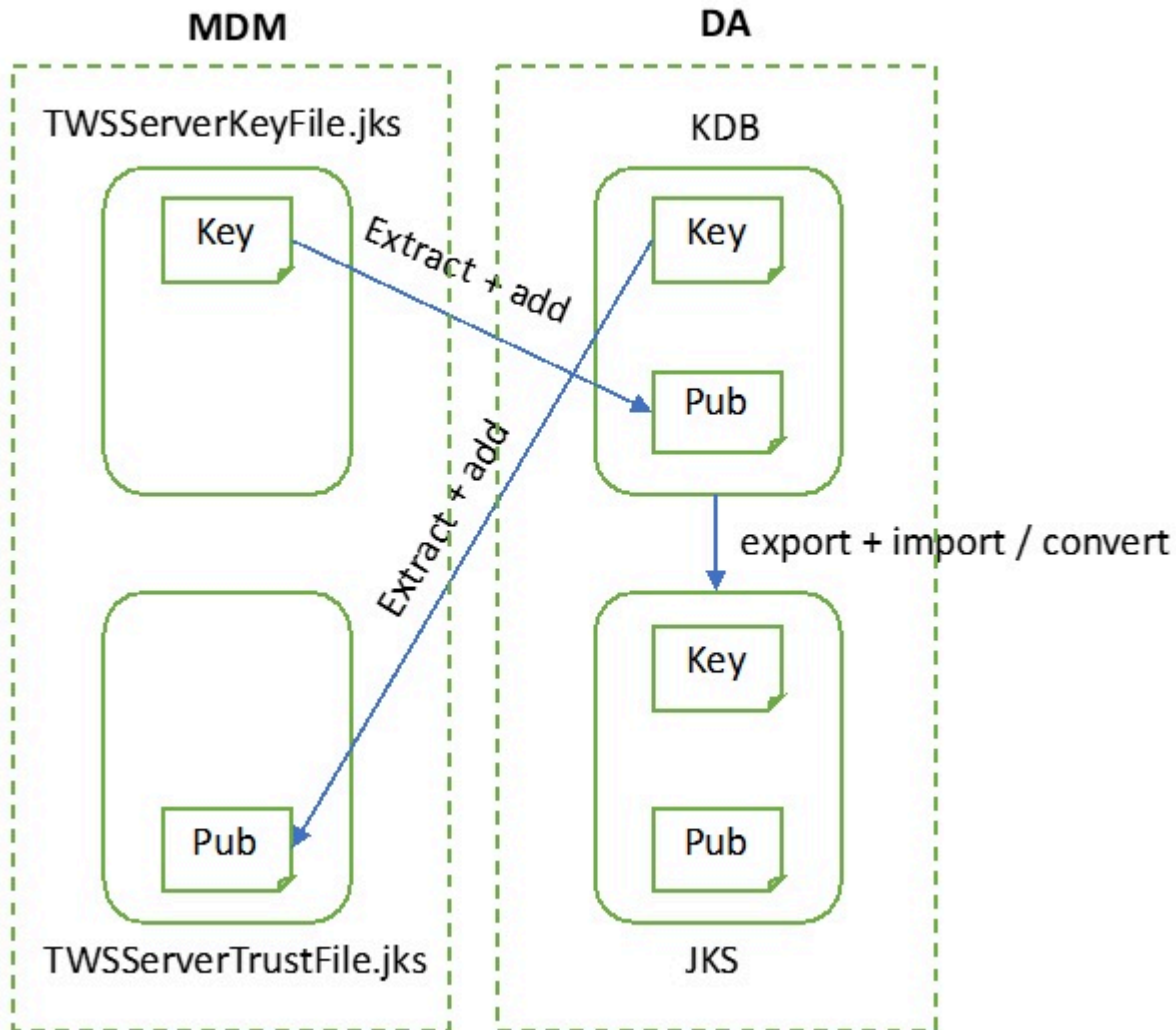
Note: Customizing certificates is a sophisticated procedure which requires a background knowledge in security, encryption, certificates, and keys management. The procedures listed in this section provide basic examples which you have to adapt to your environment and requirements.

When customizing certificates for communication between master domain manager and dynamic agent, the following scenarios are supported:

- [Customizing master domain manager and dynamic agent certificates on page 304](#)
- [Customizing master domain manager certificates on page 305](#)
- [Customizing dynamic agent certificates on page 307](#)

Figure 12: Overview of keys distribution between MDM and DWC on page 312 indicates how keys are distributed between master domain manager and dynamic agent.

Figure 10. Overview of keys distribution between master domain manager and dynamic agent



For information about how to extend these communication scenarios to backup master domain manager and dynamic domain manager, see [Extending communication scenarios to other server components on page 315](#).

Customizing master domain manager and dynamic agent certificates

Procedure to use customized certificates for communication between master domain manager and dynamic agent

About this task

The procedure explained below is one of several procedures you can perform to achieve the same results and is intended only as an example. In this procedure, it is assumed your certificates have been signed by a Certificate Authority (CA) you created for this purpose. For more information, see [Creating a Certificate Authority on page 301](#). For more information about using an external CA or manually modifying all the keystores and key databases, see [Replacing Default SSL Certificates with CA Signed Custom Certificates](#).

To customize the certificates for communication between master domain manager and dynamic agent, perform the following steps:

1. On the master domain manager, generate a self-signed certificate or issue a certificate sign request to a CA and import the certificate into `TWSServerKeyFile.jks`. For example, you can generate the private key to be used for signing the custom certificate by issuing the following command:

```
openssl genrsa -des3 -out tls.key 2048
```

2. Create the certificate sign request:

```
openssl req -new -key tls.key -out tls.csr -config
/usr/Tivoli/TWS/OpenSSL64/1.0.0/bin/openssl.cnf
```

3. Send the `.csr` to the CA:

```
openssl x509 -req -in tls.csr -days 3650
-CA ca.crt -CAkey ca.key -CAcreateserial -out tls.crt
```

4. After receiving back the signed certificate, you can import the custom certificate along with its private key into `TWSServerKeyFile.jks`, as follows:

- a. Create a single file containing both:

```
cat tls.key tls.crt > tls.tot
```

- b. Export the resulting file to a PKCS12 keystore:

```
openssl pkcs12 -export -out TWSServerKeyFile.p12 -in tls.tot -name server
```

- c. Import the PKCS12 keystore into `TWSServerKeyFile.jks`:

```
keytool -importkeystore -srckeystore TWSServerKeyFile.p12 -srcstoretype pkcs12
-destkeystore TWSServerKeyFile.jks -deststoretype jks -srcstorepass password
-deststorepass password -srcaalias server -destalias server
```

5. On the master domain manager, import the CA certificate in the path `<TWSDATA>/ssl/TWSClientKeyStoreJKS.jks` :

```
keytool -importcert -file ca.crt -keystore TWSClientKeyStoreJKS.jks
-alias ca -trustcacerts
```

6. On the master domain manager, edit the `TWA_DATA_DIR/broker/config/BrokerWorkstation.properties` file and update the list of authorized Common Names for the dynamic domain manager (broker). Append the Common Name used for the custom certificate to the `Broker.AuthorizedCNs` property:

```
Broker.AuthorizedCNs=Server;ServerNew;new_CN
```

7. Run the `AgentCertificateDownloader` script on the dynamic agent. The script connects to the master domain manager, downloads the certificates in .PEM format (`tls.key`, `tls.crt`, `ca.crt` files), and deploys them to the agent. The certificates must be available on the master domain manager in a specific path. For more information, see [Certificates download to dynamic agents - AgentCertificateDownloader script](#).

Customizing master domain manager certificates

Procedure to use customized certificates for the master domain manager

About this task

The procedure explained below is one of several procedures you can perform to achieve the same results and is intended only as an example. In this procedure, it is assumed your certificates have been signed by a Certificate Authority (CA) you created for this purpose. For more information, see [Creating a Certificate Authority on page 301](#). For more information about using an external CA or manually modifying all the keystores and key databases, see [Replacing Default SSL Certificates with CA Signed Custom Certificates](#).

To customize the master domain manager certificates, perform the following steps:

1. On the master domain manager, generate a self-signed certificate or issue a certificate sign request to a CA and import the certificate into `TWSServerKeyFile.jks`. For example, you can generate the private key to be used for signing the custom certificate by issuing the following command:

```
openssl genrsa -des3 -out tls.key 2048
```

2. Create the certificate sign request:

```
openssl req -new -key tls.key -out tls.csr -config
/usr/Tivoli/TWS/OpenSSL64/1.0.0/bin/openssl.cnf
```

3. Send the `.csr` to the CA:

```
openssl x509 -req -in tls.csr -days 3650
-CA ca.crt -CAkey ca.key -CAcreateserial -out tls.crt
```

4. After receiving back the signed certificate, you can import the custom certificate along with its private key into `TWSServerKeyFile.jks`, as follows:

- a. Create a single file containing both:

```
cat tls.key tls.crt > tls.tot
```

- b. Export the resulting file to a PKCS12 keystore:

```
openssl pkcs12 -export -out TWSServerKeyFile.p12 -in tls.tot -name server
```

- c. Import the PKCS12 keystore into `TWSServerKeyFile.jks`:

```
keytool -importkeystore -srckeystore TWSServerKeyFile.p12 -srcstoretype pkcs12
-destkeystore TWSServerKeyFile.jks -deststoretype jks -srcstorepass password
-deststorepass password -srcaalias server -destalias server
```

5. On the master domain manager, import the CA certificate in the path `<TWSDATA>/ssl/TWSClientKeyStoreJKS.jks` :

```
keytool -importcert -file ca.crt -keystore TWSClientKeyStoreJKS.jks
-alias ca -trustcacerts
```

6. On the master domain manager, extract the public key to a certificate file from the private key of the master domain manager keystore (`TWSServerKeyFile.jks`):

```
keytool -exportcert -alias server -file pkserver.cer
-keystore TWSServerKeyFile.jks -storetype jks
```

7. On the master domain manager, edit the `TWA_DATA_DIR/broker/config/BrokerWorkstation.properties` file and update the list of authorized Common Names for the dynamic domain manager (broker). Append the Common Name used for the custom certificate to the `Broker.AuthorizedCNs` property:

```
Broker.AuthorizedCNs=Server;ServerNew;new_CN
```

- On the dynamic agent, add the certificate extracted at step 6 on page 306 into the keystore of the dynamic agent `TWSSClientKeyStore.kdb` and into `TWSSClientKeyStoreJKS.jks`:

```
gsk8capicmd_64 -cert -add -db TWSSClientKeyStore.kdb
-file pkserver.cer -label server -trust enable -stashed
```

- Add the same certificate to `TWSSClientKeyStoreJKS.jks`:

```
keytool -importcert -file pkserver.cer -keystore TWSSClientKeyStoreJKS.jks
-alias server
```

Customizing dynamic agent certificates

Procedure to use customized certificates for the dynamic agent

About this task

To customize dynamic agent certificates, perform the following steps:

- On the master domain manager, generate a self-signed certificate or issue a certificate sign request to a CA. For example, you can generate the private key to be used for signing the custom certificate by issuing the following command:

```
openssl genrsa -des3 -out tls.key 2048
```

- Create the certificate sign request:

```
openssl req -new -key tls.key -out tls.csr -config
/usr/Tivoli/TWS/OpenSSL64/1.0.0/bin/openssl.cnf
```

- Send the `.csr` to the CA:

```
openssl x509 -req -in tls.csr -days 3650
-CA ca.crt -CAkey ca.key -CAcreateserial -out tls.crt
```

- Run the `AgentCertificateDownloader` script on the dynamic agent. The script connects to the master domain manager, downloads the certificates in `.PEM` format (`tls.key`, `tls.crt`, `ca.crt` files), and deploys them to the agent. The certificates must be available on the master domain manager in a specific path. For more information, see [Certificates download to dynamic agents - AgentCertificateDownloader script](#).

- On the master domain manager, import the CA certificate in the path `<TWSDATA>/ssl/`

`TWSSClientKeyStoreJKS.jks` :

```
keytool -importcert -file ca.crt -keystore TWSSClientKeyStoreJKS.jks
-alias ca -trustcacerts
```

- On the master domain manager, extract the public key to a certificate file from the private key of the master domain manager keystore (`TWSServerKeyFile.jks`):

```
keytool -exportcert -alias server -file pkserver.cer
-keystore TWSServerKeyFile.jks -storetype jks
```

- On the master domain manager, edit the `TWA_DATA_DIR/broker/config/BrokerWorkstation.properties` file and update the list of authorized Common Names for the dynamic domain manager (broker). Append the Common Name used for the custom certificate to the `Broker.AuthorizedCNs` property:

```
Broker.AuthorizedCNs=Server;ServerNew;new_CN
```

8. On the dynamic agent, add the certificate extracted at step 6 into the keystore of the dynamic agent

`TWSCliantKeyStore.kdb` and into `TWSCliantKeyStoreJKS.jks`:

```
gsk8scapicmd_64 -cert -add -db TWSCliantKeyStore.kdb
-file pkserver.cer -label server -trust enable -stashed
```

9. Add the same certificate to `TWSCliantKeyStoreJKS.jks`:

```
keytool -importcert -file pkserver.cer -keystore TWSCliantKeyStoreJKS.jks
-alias server
```

Configuring custom certificates for the remote broker resource CLI

About this task

To use custom certificates on agents, perform the following steps:

In `TWA_DATA_DIR/TDWB_CLI/config/CLIConfig.properties` and `TWA_DATA_DIR/broker/config/CLIConfig.properties`, customize the following properties with the directory where your custom `.JKS` files are stored and the associated passwords:

- KeyStore and trustStore files:
 - `keyStore=<TWA_DATA_DIR>/ITA/cpa/ita/cert/<JKS_ClientKeyStoreFile>`
 - `trustStore=<TWA_DATA_DIR>/ITA/cpa/ita/cert/<JKS_ClientTrustStoreFile>`
- KeyStore and trustStore passwords:
 - `keyStorepwd=<customPassword>`
 - `trustStorepwd=<customPassword>`

Scenario: Connection between the Dynamic Workload Console and the IBM Workload Scheduler components

The Dynamic Workload Console connects in SSL mode with the IBM Workload Scheduler components by using the default certificates. You might configure the Dynamic Workload Console to connect in SSL mode by using your certificates.

You can have SSL communication between the Dynamic Workload Console and one of the following components:

- master domain manager
- Backup master domain manager
- Dynamic domain manager
- Backup dynamic domain manager
- Agent

When you are customizing the Dynamic Workload Console settings, make sure the keys have the same password as the keystore where they are saved. The Dynamic Workload Console keystore password must be the same as the Dynamic Workload Console client and IBM Workload Scheduler server.



Note: When you configure the Dynamic Workload Console to connect to different agents, the Dynamic Workload Console truststore must have a certificate for each agent to enable SSL connection.

Overview

For more information about the SSL connection between Dynamic Workload Console and IBM Workload Scheduler components, see [Overview on page 309](#).

SSL connection by using default certificates

For more information about the SSL default connection, see [SSL connection by using the default certificates on page 302](#).

SSL connection by using your certificates

For more information about how to create and enable your SSL certificates, see [Customizing certificates for master domain manager and Dynamic Workload Console communication on page 311](#).

Overview

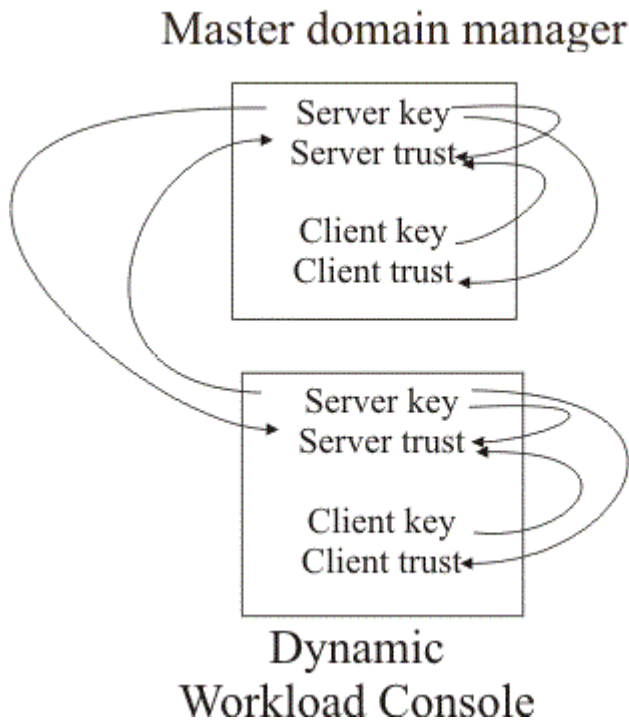
Overview of the Dynamic Workload Console SSL connection

To implement the RMI/IIOP over SSL communication between the Dynamic Workload Console and the internal communication of master domain manager, backup master domain manager, dynamic domain manager, backup dynamic domain manager or agent, you use the server and client security features of WebSphere Application Server Liberty Base.

The SSL security paradigm implemented in the WebSphere Application Server Liberty Base requires two stores to be present on the clients and the server: a keystore containing the private key and a truststore containing the certificates of the trusted counterparts.

[Figure 11: SSL server and client keys on page 310](#) shows the server and client keys, and to where they must be exported for the Dynamic Workload Console:

Figure 11. SSL server and client keys



The diagram shows the keys Dynamic Workload Console and components must extract and distributed to enable SSL communication. The Dynamic Workload Console interface uses the default certificates that are installed in the default keystores to communicate with the agent. You can configure the Dynamic Workload Console to connect in SSL mode with an agent by using your certificates to meet your required security settings.

In addition creating new keys, you can also customize the name, location, and password of the keystore and truststore. For details about possibilities, see [Table 64: Changes allowed in IBM Workload Scheduler keystore and truststore on page 310](#).

Table 64. Changes allowed in IBM Workload Scheduler keystore and truststore

File	Name	Path	Password	New key
TWS server keystore	✓	✓	✓	✓
TWS server truststore	✓	✓	✓	✓
TWS client keystore				✓
TWS client truststore				✓
TDWC client keystore				✓
TDWC client truststore				✓

When you are customizing the Dynamic Workload Console settings, make sure that the keys have the same password as the keystore where they are saved. The Dynamic Workload Console keystore password must be the same as the Dynamic Workload Console client and IBM Workload Scheduler server password.



Note: When you configure the Dynamic Workload Console to connect to different agents, the Dynamic Workload Console truststore must have a certificate for each component to enable SSL connection.

Customizing certificates for master domain manager and Dynamic Workload Console communication

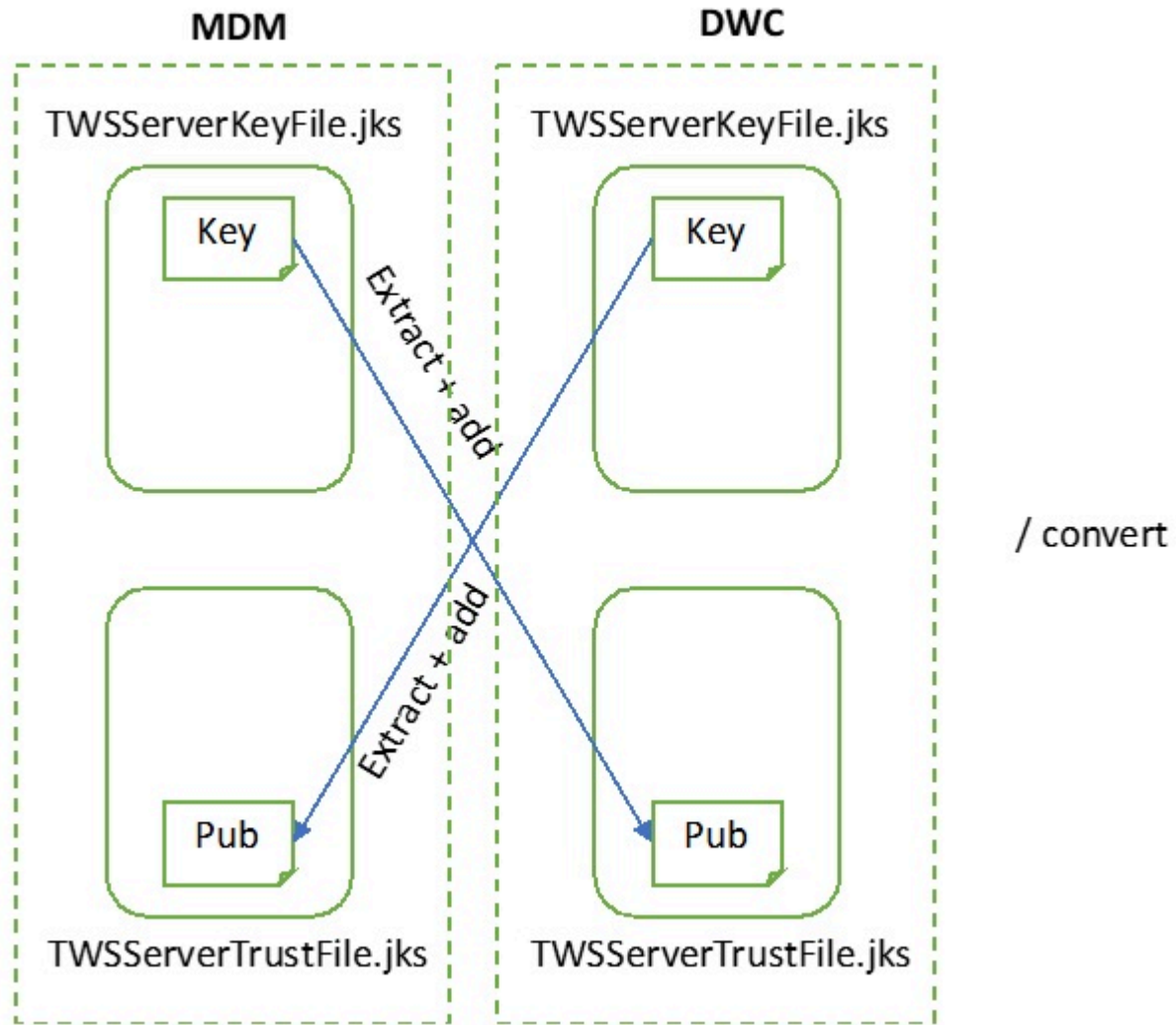
Supported scenarios for creating custom certificates for communication between master domain manager and Dynamic Workload Console

When customizing certificates for communication between master domain manager and Dynamic Workload Console, the following scenarios are supported::

- [Customizing master domain manager and Dynamic Workload Console certificates on page 312](#)
- [Customizing master domain manager certificates on page 313](#)
- [Customizing Dynamic Workload Console certificates on page 314](#)

[Figure 12: Overview of keys distribution between MDM and DWC on page 312](#) indicates how keys are distributed between master domain manager and Dynamic Workload Console.

Figure 12. Overview of keys distribution between MDM and DWC



For information about how to extend these communication scenarios to backup master domain manager and dynamic domain manager, see [Extending communication scenarios to other server components on page 315](#).

Customizing master domain manager and Dynamic Workload Console certificates

Supported scenarios for creating custom certificates for communication between master domain manager and Dynamic Workload Console

About this task

To customize the certificates for communication between master domain manager and Dynamic Workload Console, perform the following steps:

1. On the master domain manager, generate a self-signed certificate or issue a certificate sign request to a CA and import the certificate into `TWSServerKeyFile.jks`. For example, you can generate the private key to be used for signing the custom certificate by issuing the following command:

```
openssl genrsa -des3 -out tls.key 2048
```

2. Create the certificate sign request:

```
openssl req -new -key tls.key -out tls.csr -config
/usr/Tivoli/TWS/OpenSSL64/1.0.0/bin/openssl.cnf
```

3. After receiving back the signed certificate, you can import the custom certificate along with its private key into `TWSServerKeyFile.jks`, as follows:

- a. Create a single file containing both:

```
cat tls.key tls.crt > tls.tot
```

- b. Export the resulting file to a PKCS12 keystore:

```
openssl pkcs12 -export -out TWSServerKeyFile.p12 -in tls.tot -name server
```

- c. Import the PKCS12 keystore into `TWSServerKeyFile.jks`:

```
keytool -importkeystore -srckeystore TWSServerKeyFile.p12 -srcstoretype pkcs12
-destkeystore TWSServerKeyFile.jks -deststoretype jks -srcstorepass password
-deststorepass password -srcaalias server -destalias server
```

4. On the master domain manager, import the CA certificate in the path `<TWSDATA>/ssl/TWSClientKeyStoreJKS.jks` :

```
keytool -importcert -file ca.crt -keystore TWSClientKeyStoreJKS.jks
-alias ca -trustcacerts
```

5. Copy the `TWSServerKeyFile.jks` and `TWSServerTrustFile.jks` files from the master domain manager to the Dynamic Workload Console, overwriting the existing files.

Customizing master domain manager certificates

Procedure to use custom certificates for the master domain manager

About this task

To customize the master domain manager certificates, perform the following steps:

1. On the master domain manager, generate a self-signed certificate or issue a certificate sign request to a CA and import the certificate into `TWSServerKeyFile.jks`. For example, you can generate the private key to be used for signing the custom certificate by issuing the following command:

```
openssl genrsa -des3 -out tls.key 2048
```

2. Create the certificate sign request:

```
openssl req -new -key tls.key -out tls.csr -config
/usr/Tivoli/TWS/OpenSSL64/1.0.0/bin/openssl.cnf
```

3. After receiving back the signed certificate, you can import the custom certificate along with its private key into `TWSServerKeyFile.jks`, as follows:

- a. Create a single file containing both:

```
cat tls.key tls.crt > tls.tot
```

- b. Export the resulting file to a PKCS12 keystore:

```
openssl pkcs12 -export -out TWSServerKeyFile.p12 -in tls.tot -name server
```

- c. Import the PKCS12 keystore into TWSServerKeyFile.jks:

```
keytool -importkeystore -srckeystore TWSServerKeyFile.p12 -srcstoretype pkcs12
-destkeystore TWSServerKeyFile.jks -deststoretype jks -srcstorepass password
-deststorepass password -srcaalias server -destalias server
```

4. On the master domain manager, import the CA certificate in the path <TWSDATA>/ssl/

TWSSClientKeyStoreJKS.jks :

```
keytool -importcert -file ca.crt -keystore TWSSClientKeyStoreJKS.jks
-alias ca -trustcacerts
```

5. On the Dynamic Workload Console, import the CA certificate into the TWSServerTrustFile.jks:

```
keytool -importcert -file ca.crt -keystore TWSServerTrustFile.jks -alias ca
-trustcacerts
```

Customizing Dynamic Workload Console certificates

Procedure to use custom certificates for the Dynamic Workload Console

About this task

To customize the Dynamic Workload Console certificates, perform the following steps:

1. On the master domain manager, generate a self-signed certificate or issue a certificate sign request to a CA and import the certificate into TWSServerKeyFile.jks. For example, you can generate the private key to be used for signing the custom certificate by issuing the following command:

```
openssl genrsa -des3 -out tls.key 2048
```

2. Create the certificate sign request:

```
openssl req -new -key tls.key -out tls.csr -config
/usr/Tivoli/TWS/OpenSSL64/1.0.0/bin/openssl.cnf
```

3. After receiving back the signed certificate, you can import the custom certificate along with its private key into TWSServerKeyFile.jks, as follows:

- a. Create a single file containing both:

```
cat tls.key tls.crt > tls.tot
```

- b. Export the resulting file to a PKCS12 keystore:

```
openssl pkcs12 -export -out TWSServerKeyFile.p12 -in tls.tot -name server
```

- c. Import the PKCS12 keystore into TWSServerKeyFile.jks:

```
keytool -importkeystore -srckeystore TWSServerKeyFile.p12 -srcstoretype pkcs12
-destkeystore TWSServerKeyFile.jks -deststoretype jks -srcstorepass password
-deststorepass password -srcaalias server -destalias server
```

4. On the Dynamic Workload Console, import the A certificate into `TWSServerTrustFile.jks`:

```
keytool -importcert -file ca.crt -keystore TWSServerTrustFile.jks
-alias ca -trustcacerts
```

5. On the master domain manager, import the CA certificate in the path `<TWSDATA>/ssl/`

`TWSClientKeyStoreJKS.jks` :

```
keytool -importcert -file ca.crt -keystore TWSClientKeyStoreJKS.jks
-alias ca -trustcacerts
```

Extending communication scenarios to other server components

Apply custom certificates to backup master domain manager and dynamic domain manager

You can extend the certificate customization scenarios explained in [Customizing certificates for master domain manager and dynamic agent communication on page 303](#) and [Customizing certificates for master domain manager and Dynamic Workload Console communication on page 311](#) also to backup master domain manager and dynamic domain manager keeping in mind the following points:

- The backup master domain manager can communicate with Dynamic Workload Console, dynamic agent, and fault-tolerant agent.

In all communication scenarios, the backup master domain manager assumes the role of master domain manager, so the procedures described above in [Customizing certificates for master domain manager and dynamic agent communication on page 303](#) and [Customizing certificates for master domain manager and Dynamic Workload Console communication on page 311](#) must be applied by replacing the master domain manager component with the backup master domain manager.

It is recommended to replicate on the backup master domain manager all key changes performed on the master domain manager. This can be obtained by copying the keystores or exporting and importing the keys. Keystores for the backup master domain manager have the same names and location as the master domain manager ones.

- The dynamic domain manager can communicate with the Dynamic Workload Console and dynamic agent.

In all communication scenarios, the dynamic domain manager assumes the role of master domain manager, so the procedures described above in [Customizing certificates for master domain manager and dynamic agent communication on page 303](#) and [Customizing certificates for master domain manager and Dynamic Workload Console communication on page 311](#) must be applied by replacing the master domain manager component with the dynamic domain manager.

Keystores for the dynamic domain manager have the same names and location as the master domain manager ones.

- The dynamic domain manager can communicate with the master domain manager. This is a two-way communication between two servers, so the certificate customization procedure is the same as explained in [Customizing certificates for master domain manager and Dynamic Workload Console communication on page 311](#), where the dynamic domain manager assumes the role of the Dynamic Workload Console. Keystores for the dynamic domain manager have the same names and location as the master domain manager ones.

Creating new brand keystores

Creating new brand keystores for the server components and dynamic agent

In addition to the customization scenarios described in [Customizing certificates for master domain manager and dynamic agent communication on page 303](#) and [Customizing certificates for master domain manager and Dynamic Workload Console communication on page 311](#), you can optionally create brand new keystores for both the server components and the dynamic agent, as follows:

- [Creating brand new keystores for the server components on page 316](#)
- [Creating brand new keystores for the dynamic agent on page 316](#)

Creating brand new keystores for the server components

Create brand new keystores for the master domain manager (backup master domain manager, dynamic domain manager) and Dynamic Workload Console

About this task

To create new keystores for the master domain manager (backup master domain manager, dynamic domain manager) and Dynamic Workload Console, perform the following steps:

1. Create one keystore for the private keys and one keystore for the trusted keys in the .jks format and save them in the same folders as the old files.
2. Browse to the `ssl_variables.xml` file, which is located in

On UNIX operating systems

```
TWA_DATA_DIR/usr/servers/engineServer/configDropins/overrides
```

On Windows operating systems

```
TWA_home\usr\servers\engineServer\configDropins\overrides
```

3. Update the following variables in the `ssl_variables.xml` file with the new values:
 - `keyStore.location`
 - `keyStore.password`
 - `trustStore.location`
 - `trustStore.password`
4. To verify that the certificate customization has completed successfully, delete the old certificates and optionally rename the new certificates with the names of the certificates you deleted.

Creating brand new keystores for the dynamic agent

Create brand new keystores for the dynamic agent

About this task

To create brand new keystores for the dynamic agent, perform the following steps:

1. On the dynamic agent, create a CMS (.kdb) keystore.
2. On the dynamic agent, use the **twsmakekey** script to convert the keystore created in step 1.
3. Stop the dynamic agent using the **ShutdownLwa** command.
4. Browse to the `ita.ini` file, which is located in the following paths:

On Windows systems

`<TWA_home>\TWS\ITA\cpa\ita`

On UNIX systems

`<TWA_DATA_DIR>/ITA/cpa/ita`

5. Edit the following properties in the `ita.ini` file:

key_db_name

file name of the .kdb generated in step 1 (CMS)

key_repository_dir

path to the .kdb generated in step 1

java_truststore_password_file

absolute path to the .sth generated in step 2 (pwd = default)

java_truststore_file

absolute path to the .jks generated in step 2

6. Restart the dynamic agent using the **StartupLwa** command.
7. To verify that the certificate customization has completed successfully, delete the old certificates and optionally rename the new certificates with the names of the certificates you deleted.

Scenario: SSL Communication across the fault-tolerant agent network

You can enable the SSL connection using OpenSSL Toolkit for the following components:

- Master domain manager and its domain managers
- Master domain manager and fault-tolerant agents in the master domain
- Master domain manager and backup master domain manager
- Domain manager and fault-tolerant agents that belong to that domain

The default certificates are located in the `<TWA_HOME>\TWS\ssl\OpenSSL` directory.

Using SSL for netman and conman

IBM Workload Scheduler provides a secure, authenticated, and encrypted connection mechanism for communication across the network topology. This mechanism is based on the Secure Sockets Layer (SSL) protocol and uses the OpenSSL Toolkit, which is automatically installed with IBM Workload Scheduler.

The SSL protocol is based on a private and public key methodology. SSL provides the following authentication methods:

CA trusting only

Two workstations trust each other if each receives from the other a certificate that is signed or is trusted. That is, if the CA certificate is in the list of trusted CAs on each workstation. With this authentication level, a workstation does not perform any additional checks on certificate content, such as the distinguished name. Any signed or trusted certificate can be used to establish an SSL session. See [Setting local options on page 51](#) for a definition of the `caonly` option used by the `ssl_auth_mode` keyword.

Check if the distinguished name matches a defined string

Two workstations trust each other if, after receiving a trusted or signed certificate, each performs a further check by extracting the distinguished name from the certificate and comparing it with a string that was defined in its local options file. See [Setting local options on page 51](#) for a definition of the string option.

Check if the distinguished name matches the workstation name

Two workstations trust each other if, after receiving a signed or trusted certificate, each performs a further check by extracting the distinguished name from the certificate and comparing it with the unique ID of the workstation that sent the certificate. You can obtain the unique ID by using the `;showid` composer filter. Only if the unique ID is empty, you can use the name of the workstation instead of the unique ID.

See [Setting local options on page 51](#) for a definition of the `cpu` option.

To provide SSL security for a domain manager attached to z/OS® in an end-to-end connection, configure the OS/390® Cryptographic Services System SSL in the IBM Workload Scheduler code that runs in the OS/390® USS UNIX® shell in the IBM Z Workload Scheduler server address space. See the IBM Z Workload Scheduler documentation.

When configuring SSL you can:

Use the same certificate for the entire network

If the workstations are configured with CA trusting only, they accept connections with any other workstation that sends a signed or trusted certificate. To enforce the authentication you define a name or a list of names that must match the contents of the certificate distinguished name (DN) field in the `localopts` file of each workstation.

Use a certificate for each domain

Install private keys and signed certificates for each domain in the network. Then, configure each workstation to accept a connection only with partners that have a particular string of the certificate DN field in the `localopts` file of each workstation.

Use a certificate for each workstation

Install a different key and a signed certificate on each workstation and add a Trusted CA list containing the CA that signed the certificate. Then, configure each workstation to accept a connection only with partners that have their workstation name specified in the `Symphony` file recorded in the DN field of the certificate.

Setting up private keys and certificates

About this task

To use SSL authentication on a workstation, you need to create and install the following:

- The private key and the corresponding certificate that identify the workstation in an SSL session.
- The list of certificate authorities that can be trusted by the workstation.

Use the **openssl** command line utility to:

- Create a file containing pseudo random generated bytes (TWS.rnd). This file is needed on some operating systems for SSL to function correctly.
- Create a private key.
- Save the password you used to create the key into a file.
- Create a Certificate Signing Request.
- Send this Certificate Signing Request (CSR) to a Certifying Authority (CA) for signing, or:
 - Create your own Certificate Authority (CA)
 - Create a self-signed CA Certificate (X.509 structure) with the RSA key of your own CA
 - Use your own Certificate Authority (CA) to sign and create real certificates

These actions will produce the following files that you will install on the workstation(s):

- A private key file (for example, TWS.key). This file should be protected, so that it is not stolen to use the workstation's identity. You should save it in a directory that allows read access to the TWS user of the workstation, such as *TWA_home/TWS/ssl/TWS.key*.
- The corresponding certificate file (for example, TWS.crt). You should save it in a directory that allows read access to the TWS user of the workstation, such as *TWA_home/TWS/ssl/TWS.crt*.
- A file containing a pseudo-random generated sequence of bytes. You can save it in any directory that allows read access to the TWS user of the workstation, such as *TWA_home/TWS/ssl/TWS.rnd*.

In addition, you should create the following:

- A file containing the password used to encrypt the private key. You should save it in a directory that allows read access to the TWS user of the workstation, such as *TWA_home/TWS/ssl/TWS.sth*.
- The certificate chain file. It contains the concatenation of the PEM-encoded certificates of certification authorities which form the certificate chain of the workstation's certificate. This starts with the issuing CA certificate of the workstation's certificate and can range up to the root CA certificate. Such a file is simply the concatenation of the various PEM-encoded CA certificate files, usually in certificate chain order.
- The trusted CAs file. It contains the trusted CA certificates to use during authentication. The CAs in this file are also used to build the list of acceptable client CAs passed to the client when the server side of the connection requests a client certificate. This file is simply the concatenation of the various PEM-encoded CA certificate files, in order of preference.

Creating Your Own Certification Authority

About this task

If you are going to use SSL authentication within your company's boundaries and not for outside internet commerce, you might find it simpler to create your own certification authority (CA) to trust all your IBM Workload Scheduler installations. To do so, follow the steps listed below.



Note: In the following steps, the names of the files created during the process TWS and TWScA are sample names. You can use your own names, but keep the same file extensions.

1. Choose a workstation as your CA root installation.
2. Type the following command from the SSL directory to initialize the pseudo random number generator, otherwise subsequent commands might not work.

- On UNIX™:

```
$ openssl rand -out TWS.rnd -rand ./openssl 8192
```

- On Windows™:

```
$ openssl rand -out TWS.rnd -rand ./openssl.exe 8192
```

3. Type the following command to create the CA private key:

```
$ openssl genrsa -out TWScA.key 2048
```

4. Type the following command to create a self-signed CA Certificate (X.509 structure):

```
$ openssl req -new -x509 -days 365 -key TWScA.key -out TWScA.crt -config ./openssl.cnf
```

Now you have a certification authority that you can use to trust all of your installations. If you want, you can create more than one CA.

Creating private keys and certificates

About this task

The following steps explain how to create one key and one certificate. You can decide to use one key and certificate pair for the entire network, one for each domain, or one for each workstation. The steps below assume that you will be creating a key and certificate pair for each workstation and thus the name of the output files created during the process has been generalized to *workstationname*.

On each workstation, perform the following steps to create a private key and a certificate:

1. Enter the following command from the SSL directory to initialize the pseudo random number generator, otherwise subsequent commands might not work.

- On Windows™ operating systems:

```
$ openssl rand -out workstationname.rnd -rand ./openssl.exe 8192
```

- On UNIX™ and Linux™ operating systems :

```
$ openssl rand -out workstationname.rnd -rand ./openssl 8192
```

2. Enter the following command to create the private key (this example shows triple-DES encryption):


```
$ openssl genrsa -des3 -out workstationname.key 2048
```

Then, save the password that was requested to encrypt the key in a file named *workstationname.pwd*.



Note: Verify that file *workstationname.pwd* contains just the characters in the password. For instance, if you specified the word *maestro* as the password, your *workstationname.pwd* file should not contain any CR or LF characters at the end (it should be 7 bytes long).

3. Create stash file, you can choose to create a stash file or encrypt you password file:

stash file

Enter the following command to save your password, encoding it in base64 into the appropriate stash file:

```
$ openssl base64 -in workstationname.pwd -out workstationname.sth
```

You can then delete file *workstationname.pwd*.

encrypted password file

Run the following command to save your encrypted password, encoding it in base64:

```
$ conman crypt workstationname.pwd
```

Example: If you have the *workstationname.pwd* that contains the string *secret* that is the password you set, after you run the `$ conman crypt workstationname.pwd`, your *workstationname.pwd* file contains the string `{3DES}poh56FeTy+=/jhtf2djur` that is the encrypted password.

4. Enter the following command to create a certificate signing request (CSR):

```
$ openssl req -new -key workstationname.key -out workstationname.csr
-config ./openssl.cnf
```

Some values—such as company name, personal name, and more—will be requested at screen. For future compatibility, you might specify the workstation name as the distinguished name.

5. Send the *workstationname.csr* file to your CA in order to get the matching certificate for this private key.

Using its private key (*TWSca.key*) and certificate (*TWSca.crt*), the CA will sign the CSR (*workstationname.csr*) and create a signed certificate (*workstationname.crt*) with the following command:

```
$ openssl x509 -req -CA TWSca.crt -CAkey TWSca.key -days 365
-in workstationname.csr -out workstationname.crt -CAcreateserial
```

6. Distribute to the workstation the new certificate *workstationname.crt* and the public CA certificate *TWSca.crt*.

The table below summarizes which of the files created during the process have to be set as values for the workstation's local options.

Table 65. Files for Local Options

Local option	File
SSL key	<i>workstationname.key</i>
SSL certificate	<i>workstationname.crt</i>
SSL key pwd	<i>workstationname.sth</i>
SSL ca certificate	TWSca.crt
SSL random seed	<i>workstationname.rnd</i>

Configuring SSL attributes

Use the composer command line or the Dynamic Workload Console to update the workstation definition in the database. See the *IBM Workload Scheduler: User's Guide and Reference* for further information.

Configure the following attributes:

secureaddr

Defines the port used to listen for incoming SSL connections. This value must match the one defined in the **nm SSL port** local option of the workstation. It must be different from the **nm port** local option that defines the port used for normal communications. If **securitylevel** is specified but this attribute is missing, 31113 is used as the default value.

securitylevel

Specifies the type of SSL authentication for the workstation. It must have one of the following values:

enabled

The workstation uses SSL authentication only if its domain manager workstation or another fault-tolerant agent below it in the domain hierarchy requires it.

on

The workstation uses SSL authentication when it connects with its domain manager. The domain manager uses SSL authentication when it connects to its parent domain manager. The fault-tolerant agent refuses any incoming connection from its domain manager if it is not an SSL connection.

force

The workstation uses SSL authentication for all of its connections and accepts connections from both parent and subordinate domain managers. It will refuse any incoming connection if it is not an SSL connection.

If this attribute is omitted, the workstation is not configured for SSL connections. In this case, any value for **secureaddr** will be ignored. You should also set the **nm ssl port** local option to 0 to be sure that this port is not opened by netman. The following table describes the type of communication used for each type of **securitylevel** setting.

Table 66. Type of communication depending on the securitylevel value

Fault-tolerant agent (domain manager)	Domain manager (parent domain manager)	Connection type
-	-	TCP/IP
Enabled	-	TCP/IP
On	-	No connection
Force	-	No connection
-	On	TCP/IP
Enabled	On	TCP/IP
On	On	SSL
Force	On	SSL
-	Enabled	TCP/IP
Enabled	Enabled	TCP/IP
On	Enabled	SSL
Force	Enabled	SSL
-	Force	No connection
Enabled	Force	SSL
On	Force	SSL
Force	Force	SSL

The following example shows a workstation definition that includes the security attributes:

```
cpuname MYWIN
os WNT
node apollo
tcpaddr 30112
secureaddr 32222
for maestro
autolink off
fullstatus on
securitylevel on
end
```

Configuring the SSL connection protocol for the network

About this task

To configure SSL for your network, perform the following steps:

1. Create an SSL directory under the *TWA_home/TWS* directory. By default, the path *DATA_DIR/ssl* is registered in the *localopts* file. If you create a directory with a name different from *ssl* in the *DATA_DIR* directory, then update the *localopts* file accordingly. For example, if you decide to use the *TWA_home/TWS/ssl/CustomSSL/* folder instead of the default one, you can modify *localopts* as follows:

```
SSL key      ="TWA_HOME/TWS/ssl/CustomSSL/workstationname.key"
SSL certificate  ="TWA_HOME/TWS/ssl/CustomSSL/workstationname.crt"
SSL key pwd    ="TWA_HOME/TWS/ssl/CustomSSL/workstationname.sth"
SSL CA certificate  ="TWA_HOME/TWS/ssl/CustomSSL/TWSTrustCertificates.cer"
SSL random seed  ="TWA_HOME/TWS/ssl/CustomSSL/workstationname.rnd"
SSL Encryption Cipher  =HIGH
```

If you created multiple *TWScrt.crt*, you can simply append the content of each of them on a new line of the *TWSTrustCertificates.cer*.

2. Copy *openssl.cnf* and *openssl.exe* to the SSL directory.
3. Create as many private keys, certificates, and trusted CA lists as you plan to use in your network. For more information, see [Creating private keys and certificates on page 320](#).
4. For each workstation that will use SSL authentication:
 - Update its definition in the IBM Workload Scheduler database with the SSL attributes. For more information, see [Configuring SSL attributes on page 322](#).
 - Add the SSL local options in the *localopts* file.
 - Update the **SSL port** parameter. The value must match the value added to the corresponding definition in the IBM® Workload Scheduler database:

```
# Netman SSL port
# the value "0" means port close
#
nm SSL port    =PORT_NUMBER
```

For more information, see [Setting up full SSL security on page 325](#).

Although you are not required to follow a particular sequence, these tasks must all be completed to activate SSL support.

In IBM Workload Scheduler, SSL support is available for the fault-tolerant agents only (including the master domain manager and the domain managers), but not for the extended agents. If you want to use SSL authentication for a workstation that runs an extended agent, you must specify this parameter in the definition of the host workstation of the extended agent.

Configuring full SSL security

This section describes how to implement full SSL security when using an SSL connection for communication across the network by *netman* and *conman*. It contains the following topics:

- [Overview on page 325](#)
- [Setting up full SSL security on page 325](#)
- [Configuring full SSL support for internetwork dependencies on page 326](#)



Note: The full SSL security feature is not applicable to the communication between dynamic agents and the broker workstation that is defined for the master domain manager or the dynamic domain manager to which the dynamic agents are connected.

Overview

This feature provides the option to set a higher degree of SSL-based connection security on IBM Workload Scheduler networks in addition to the already available level of SSL security.

If you require a more complete degree of SSL protection, this feature supplies new configuration options to setup advanced connection security, otherwise you can use the standard settings documented above in this chapter.

The Full SSL security enhancements

Full SSL security support provides the following enhancements:

- TCP ports that can become security breaches are no longer left open.
- Traveling data, including communication headers and trailers, is now *totally* encrypted.

Compatibility between SSL support levels

The non-full and the full SSL support levels are mutually exclusive. That is, they cannot be configured simultaneously and cannot be enabled at the same time. If you enable full SSL support for an IBM Workload Scheduler network, any connection attempts by agents that are not configured for full SSL will be rejected by agents with full SSL support enabled. Vice versa, agents configured for full SSL support cannot communicate with the rest of a network set up for non-full SSL support.

Setting up full SSL security

About this task

To set full SSL connection security for your network, you must, *in addition to all the steps described above in [Connection security overview on page 300](#)*) configure the following options:

enSSLFullConnection (or `sf`)

Use `optman` on the master domain manager to set this global option to `yes` to enable full SSL support for the network. For more information, see [Setting global options on page 14](#).

nm SSL full port

If you defined the SSL port at installation time using the `netmansslport` parameter, no further action is required. For more information about the `netmansslport` parameter, see the section about agent and master installation parameters in *IBM Workload Scheduler: Planning and Installation*.

If you have not defined the SSL port at installation time, edit the `localopts` file on every agent of the network (including the master domain manager) to set this local option to the port number used to listen for incoming SSL connections. For more information, see [Setting local options on page 51](#). Take note of the following:

- This port number is to be defined also for the `SECUREADDR` parameter in the workstation definition of the agent.
- In a full SSL security setup, the `nm SSL port` local option is to be set to zero.
- You must stop netman (**conman shut;wait**) and restart it (**StartUp**) after making the changes in `localopts`.
- Check that the `securitylevel` parameter in the workstation definition of each workstation using SSL is set at least to *enabled*.

Other than the changed value for `secureaddr`, no other changes are required in the workstation definitions to set up this feature.

Configuring full SSL support for internetwork dependencies

About this task

The network agent that resolves internetwork dependencies requires a particular setup for full SSL support.

To enable a network agent for full SSL support:

1. Configure both the hosting and the remote fault-tolerant agents for full SSL support.
2. On the hosting fault-tolerant agent copy or move the `netmth.opts` file from the `DATA_DIR/config` to the `DATA_DIR/methods` directories and add (and configure) the following options:

SSL remote CPU

The workstation name of the remote master or fault-tolerant agent.

SSL remote full port

The port number defined for full SSL support on the remote master or fault-tolerant agent.

The local options that specify the private key and certificate on the hosting fault-tolerant agent

These are documented in the [Setting local options on page 51](#)).

Note that if the hosting fault-tolerant agent hosts more than one network agent, the `DATA_DIR/methods` directory contains one `netmth.opts` file for every defined network agent. In this case the complete name of each `netmth.opts` file must become:

```
network-agent-name_netmth.opts
```

If the `DATA_DIR/methods` directory contains both `network-agent-name_netmth.opts` and `netmth.opts` files, only `network-agent-name_netmth.opts` is used. If multiple agents are defined and the directory contains only `netmth.opts`, this file is used for all the network agents.

The following example adds full SSL support to the example described in *A sample network agent definition* in *User's Guide and Reference*:

- This is the workstation definition for the `NETAGT` network agent:

```
CPUNAME NETAGT
DESCRIPTION "NETWORK AGENT"
OS OTHER
NODE MASTERA.ROME.ITALY.COM
TCPADDR 31117
FOR maestro
HOST MASTERB
ACCESS NETMTH
END
```

- These are the full SSL security options in the `netmeth.opts` file of `NETAGT`:

```
#####
# Remote cpu parameters
#####

SSL remote full port = 31119
SSL remote CPU = MASTERA

#####
# Configuration Certificate
#####

SSL key           = "C:\TWS\installations\SSL\XA.key"
SSL certificate   = "C:\TWS\installations\SSL\XA.crt"
SSL CA certificate = "C:\TWS\installations\SSL\VeriSte.crt"
SSL key pwd      = "C:\TWS\installations\SSL\XA.sth"
SSL certificate chain = "C:\TWS\installations\SSL\TWSCertificateChain.crt"
SSL random seed   = "C:\TWS\installations\SSL\random_file.rnd"
SSL auth mode     = cpu
SSL auth string   = tws
```



Note: The SSL configuration certificate options must refer to the private key and certificate defined on the hosting fault-tolerant agent.

- This is the workstation definition for `MASTERA` (the remote workstation):

```
CPUNAME MASTERA
OS WNT
NODE 9.168.68.55 TCPADDR 31117
SECUREADDR 31119
DOMAIN NTWKA
FOR MAESTRO
TYPE MANAGER
AUTOLINK ON
BEHINDFIREWALL OFF
SECURITYLEVEL enabled
FULLSTATUS ON
SERVER H
END
```

Command Reference

List of commands for managing certificates

This reference section lists the commands necessary for managing certificates.

To manage certificates in JKS keystores, use the Java **keytool** command line:

```
installation_directory/JavaExt/jre/bin/keytool
```

To manage CMS (.kdb) keystore certificates, use the GSKIT command line: **gsk8capicmd**. To run the GSKIT command line, first source the TWA environment from the installation directory, as follows:

On Windows systems

```
twa_env.cmd
```

On UNIX systems

```
./twa_env.sh
```

To import a certificate, run the following command:

keytool

```
<keytool> -importkeystore -srckeystore <source keystore> -destkeystore <destination keystore>
-srcalias <certificate name in source keystore> -destalias <desired name of the certificate in
destination keystore>
-srcstorepass <password of source keystore> -deststorepass <password of destination keystore>
```

GSKIT

```
<gskit> -cert -import -db <source keystore> -pw <source keystore password>
-target <destination keystore> -target_pw <destination keystore password> -label <certificate
name>
```

To add a certificate, run the following command:

keytool

```
<keytool> -importcert -file <certificate file> -keystore <keystore name>
-alias <desired certificate name in keystore> -trustcacerts -storepass <keystore password>
```

GSKIT

```
<gskit> -cert -add -db <keystore name> -pw <keystore password>
-file <certificate file> -label <desired certificate name into keystore> -trust enable
```

To extract a certificate, run the following command:

keytool

```
<keytool> -exportcert -keystore <keystore name> -alias <name of the certificate>
-file $<file to extract into> -storepass <keystore password>
```

GSKIT

```
<gskit> -cert -extract -db <keystore name> -pw <keystore password>
-label <certificate name> -file <file to extract the certificate into>
```

To delete a certificate, run the following command:

keytool

```
<keytool> -delete -alias <certificate name> -keystore <keystore name>
-storepass <keystore password>
```

GSKIT

```
<gskit> -cert -delete -db <keystore name> -pw <keystore password>
-label <certificate name>
```

To rename a certificate, run the following command:

keytool

```
<keytool> -changealias -keystore <keystore name> -storepass <keystore password>
-alias <old certificate name> -destalias <new certificate name>
```

GSKIT

```
<gskit> -cert -rename -db <keystore name> -pw <keystore password>
-label <old certificate name> -new_label <new certificate name>
```

To list a certificate, run the following command:

keytool

```
<keytool> -list -keystore <keystore name> -storepass <keystore password>
```

GSKIT

```
<gskit> -cert -list -db <keystore name> -pw <keystore password>
```

twsManageKey script

Usage of the `twsManageKey` script

Use the `twsManageKey` command to convert kdb keystores in jks keystores.

The `twsManageKey` command is located in:

ON UNIX operating systems

```
TWA_home/TWS/_uninstall/ACTIONTOOLS
```

On Windows operating systems

```
TWA_home\TWS\_uninstall\ACTIONTOOLS
```

Syntax**On UNIX operating systems**

```
twsManageKey.sh
--certPath
--jreBinPath
--gsKitDir
--envCmd
--itaIni
--usage|-?|--help
```

On Windows operating systems

```
twsManageKey.cmd
-twsPath
-javaPath
-usage
```

Arguments

certPath

The directory where the kdb file is stored. This argument is required.

jreBinPath

The path to the jre bin location, for example: *TWA_home/JavaExt/jre/bin*. This argument is required.

gsKitDir

The path to the GSKIT location, for example: *TWA_home/TWS/GSKit64/8*, or: */usr/Tivoli/TWS/GSKit64/8*. This argument is required.

envCmd

The path to the *tws_env* command, for example: *TWA_home/TWS/tws_env.sh*. This argument is required.

itaIni

The path to the *ita.ini* file, for example: *TWA_DATA_DIR/ITA/cpa/ita/ita.ini*. This argument is required.

twsPath

The path to the *TWA_home*. This argument is required.

javaPath

The path to Java location. This argument is required.

usage

Displays the command usage

FIPS compliance

This section describes Federal Information Processing Standards (FIPS) compliance. It is divided into the following topics:

- [FIPS overview on page 331](#)
- [Using FIPS certificates on page 331](#)
- [Configuring SSL to be FIPS-compliant on page 336](#)
- [Finding the GSKit version on agents running on UNIX and Linux operating systems on page 339](#)

FIPS overview

Federal Information Processing Standards (FIPS) are standards and guidelines issued by the National Institute of Standards and Technology (NIST) for federal government computer systems. FIPS are developed when there are compelling federal government requirements for standards, such as for security and interoperability, but acceptable industry standards or solutions do not exist. Government agencies and financial institutions use these standards to ensure that the products conform to specified security requirements.

IBM Workload Automation uses cryptographic modules that are compliant with the Federal Information Processing Standard FIPS-140-2. Certificates used internally are encrypted using FIPS-approved cryptography algorithms. FIPS-approved modules can optionally be used for the transmission of data.

To satisfy the FIPS 140-2 requirement, you must use IBM Global Security Kit (GSKit) version 7d run time dynamic libraries instead of OpenSSL. GSKit uses IBM Crypto for C version 1.4.5 which is FIPS 140-2 level 1 certified by the certificate number 755. See <http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2007.htm>. IBM Java JSSE FIPS 140-2 Cryptographic is another module used by IBM Workload Automation. It has the certificate number 409.

If you are currently using SSL for secure connections across the network, to ensure FIPS compliance, you must use GSKit for secure connections instead of OpenSSL Toolkit. GSKit is automatically installed with IBM Workload Scheduler. It is based on dynamic libraries and offers several utilities for certificate management.

To comply with FIPS, all components of IBM Workload Automation must be FIPS-compliant. You must use Dynamic Workload Console or the IBM Workload Scheduler command line as the interface to IBM Workload Scheduler. Additionally, you must use DB2 as your IBM Workload Scheduler database.

If FIPS compliance is not of concern to your organization, you can continue to use SSL for secure connections across your network.

Components of IBM Workload Automation not FIPS-compliant cannot communicate with components of IBM Workload Automation FIPS-compliant.

To set FIPS compliance for your network, perform the procedures described in the following sections:

- To create FIPS certificates, see [Using FIPS certificates on page 331](#).
- To configure SSL for FIPS-compliance, see [Configuring SSL to be FIPS-compliant on page 336](#).

Using FIPS certificates

About this task

To ensure your network is FIPS-compliant, create FIPS certificates as follows:

- If you do not already have SSL certificates, see [Using fresh FIPS certificates on page 332](#).
- If you already have SSL certificates but are switching to GSKit, see [Switching from OpenSSL to GSKit on page 333](#).

If you are using FIPS certificates, you must use SSL parameters for communication over the network. During the installation or upgrade to IBM Workload Scheduler, note that default SSL certificates are located in the following directories:

```
TWA_home\TWS\ssl\GSKit
TWA_home\TWS\ssl\OpenSSL
```

Using fresh FIPS certificates

Create FIPS certificates for communication between workstations by using the `-fips` option in the GSKit command line utility. You can create FIPS certificates in the following ways:

- Use the default FIPS certificates existing on each IBM Workload Scheduler agent in the network. Note that the default FIPS certificates are not secure.
- Create your own secure FIPS certificates. See [Creating your own FIPS certificates on page 332](#).

Creating your own FIPS certificates

Use the `gsk7capicmd` command line utility to:

- Create your own Certificate Authority (CA).
- Create a self-signed CA certificate (x.509 structure) for your CA.
- Export the CA certificate in PEM format.

Creating your own Certificate Authority

Create the CA on any workstation in your network. Run the following steps only once to create a CA that will be used each time a new certificate needs to be created and signed.

1. Enter the following command to create the CMS key database "ca.kdb" with password "password00" that expires after 1000 days.

```
gsk7capicmd -keydb -create -db ca.kdb -pw password00 -stash -expire 1000 -fips
```

2. Enter the following command to create the self-signed certificate with label "CA certificate" using the distinguish name "CN=CA certificate,O=IBM,OU=TWS,C=IT". The certificate expires after 1000 days.

```
gsk7capicmd -cert -create -db ca.kdb -pw password00 -label "CA certificate"
-size 2048 -expire 1000 -dn "CN=CA certificate,O=IBM,OU=TWS,C=IT"
```

3. Enter the following command to extract the CA certificate into external file "ca.crt". The certificate is addressed by the corresponding label.

```
gsk7capicmd -cert -extract -db ca.kdb -pw password00 -label "CA certificate"
-target CA.crt
```

This file will contain the public certificate of the certificate authority.

Creating a certificate for the IBM Workload Scheduler agent

Perform the following steps to create certificates that are signed by a local common trusted CA on every IBM Workload Scheduler agent in your network.

1. Enter the following command to create a default CMS key database client.kdb" with password "password02" that expires after 1000 days. The password is also stored in stash file "client.sth".

```
gsk7capiCmd -keydb -create -db client.kdb -pw password02
             -stash -expire 1000 -fips
```

2. Enter the following command to add the CA certificate as trusted in the CMS key database. The label "CA certificate client" is used to address that certificate.

```
gsk7capiCmd -cert -add -db client.kdb -pw password02
             -label "CA certificate client" -trust enable -file CA.crt
             -format ascii -fips
```

3. Enter the following command to create the client certificate request based on 2048 bits key, with label "Client WA95 Certificate" and distinguish name "CN=Client WA95,O=IBM,OU=TWS,C=IT". The certificate request "client.csr" is generated and the private key is created in the key database client.kdb.

```
gsk7capiCmd -certreq -create -db client.kdb -pw password02
             -label "Client WA95 Certificate" -size 2048 -file client.csr
             -dn "CN=Client WA95,O=IBM,OU=TWS,C=IT" -fips
```

4. Enter the following command so that the CA signs the client's certificate request and generates a new signed in file "client.crt".

```
gsk7capiCmd -cert -sign -db ca.kdb -pw password00 -label "CA certificate"
             -target client.crt -expire 365 -file client.csr -fips
```

5. Enter the following command to import the signed certificate "client.crt" in the CMS key database "client.kdb".

```
gsk7capiCmd -cert -receive -db client.kdb -pw password02 -file client.crt -fips
```

You can repeat these steps above for all agents or you can use the same certificate for all agents, depending on your security policies and IBM Workload Scheduler localopts configurations.

Switching from OpenSSL to GSKit

About this task

This section describes how to migrate your OpenSSL certificates to GSKit certificates.

The following is a list of certificate formats that can be migrated to the GSKit format, **KDB**:

- **PEM**: Used by OpenSSL
- **JKS**: Used by Java™ and WebSphere Application Server Liberty Base
- **PKCS12**: Used by Microsoft™ applications and Internet Explorer

To migrate certificates, you may use one or more of the following tools:

- **gsk8capicmd**: Native command line provided by GSKit
- **openssl**: Native command line provided by OpenSSL
- **keytool**: Optional graphical interface provided by Java™ Virtual Machine (JVM)



Note: Be sure to backup your original certificates before migrating them to GSKit format.

To migrate your certificates, perform the following steps:

1. [Configuring the tool environment on page 334](#)
2. [Migrating the certificates on page 334](#)

Configuring the tool environment

This section describes the commands you must run to configure gsk8capicmd and openssl.

Configuring gsk8capicmd

gsk8capicmd on 32 bit

```
set PATH=C:\Program Files\IBM\TWA\TWS\GSKit32\8\lib; C:\Program Files\IBM\TWA\TWS
\GSKit32\8\bin;%PATH%
```

gsk8capicmd_64 on 64 bit

```
set PATH=C:\Program Files\IBM\TWA\TWS\GSKit64\8\lib64; C:\Program Files\IBM\TWA
\TWS\GSKit64\8\bin;%PATH%
```

Configuring openssl

UNIX™

```
twc_env.sh
```

Windows™

```
twc_env.cmd
```

Migrating the certificates

This section describes the commands you must run to migrate certificates to the FIPS-compliant format, KDB.

Note that PEM format cannot be directly converted to KDB format; you must first convert PEM to PKCS12 and then to KDB.

The following list describes the command you must run to convert from one format to another:

JKS format to KDB format

```
gsk7cmd -keydb -convert -db TWSCientKeyFile.jks -pw default -old_format jks -new_format cms
```

```
gsk7cmd -keydb -convert -db TWSCientTrustFile.kdb -pw default -old_format cms -new_format jks
```

PKCS12 format to KDB format

```
gsk7capicmd -cert -export -target TWSCientKeyFile_new.kdb -db TWSCientKeyFileP12.P12 -fips -target_type
cms -type pkcs12
```

PKCS12 format to PEM format

```
openssl pkcs12 -in TWSCientKeyFileP12.P12 -out TWSCientKeyFile.pem
```

PEM format to PKCS12 format

```
openssl pkcs12 -export -in TWSCientKeyFile.pem -out cred.p12
```

KDB format to PKCS12 format

```
gsk7capicmd -cert -export -db TWSCientKeyFile.kdb -target TWSCientKeyFileP12.P12 -fips -target_type
pkcs12 -type cms
```

Converting PEM certificates to CMS certificates

This section describes the procedure to convert PEM (OpenSSL) certificates to CMS (GSKit) certificates. The examples in this section use the following input and output files.

Input files

Personal certificate file: *CPU1.crt*
 Personal key of certificate file: *CPU1.key*
 Certificate of CA file: *TWSca.crt*
 Stash file: *CPU1.sth*

Output files

Keystore database file: *TWS.kdb*
 Stash file: *TWS.sth*
 Label of your certificate: *CPU1*

To migrate OpenSSL certificates to GSKit certificates, perform the following procedure:

1. Merge the public and private keys in a new temporary file called **all.pem** by running the following commands:

UNIX™

```
cat CPU2.crt CPU2.key > all.pem
```

Windows™

```
type CPU1.crt CPU1.key > all.pem
```

2. If you do not already know the password, extract it from the stash file by running `openssl base64 -d -in CPU1.sth`.
3. Choose a password for the new keystore database. You can reuse the old password.
4. Choose a label for your personal certificate and personal key (in this example, CPU1) and create the PKCS12 database that contains the labels. You use the name, CPU1, as the label of the new keystore database. To create the PKCS12 database, run the following:

```
openssl pkcs12 -export -in all.pem -out TWS.p12 -name CPU1 -passin pass:
password1 -passout pass:password2
```

where *password1* is the password extracted from the stash file and *password2* is the new password to manage the new keystore database.

- Convert the PKCS12 database from TWS.p12 to the CMS database, TWS.kdb by running the following:

```
gsk7capiCmd -cert -import -target TWS.kdb -db TWS.p12 -target_type cms
-type pkcs12 -label CPU1 -target_pw "password2" -pw "password3"
```

where *password2* is the old password that you extracted from the stash file, CPU1.sth and *password3* is the new password.

- Choose a label for your Certification Authority contained in TWScA.crt. For this example, it is *TWScA*.
- Add the certificate of the Certification Authority into your TWS.kdb file by running:

```
gsk7capiCmd -cert -add -db TWS.kdb -label TWScA -trust -file TWScA.crt
-format ascii -pw "password"
```

- Delete all .pem files.

Configuring SSL to be FIPS-compliant

About this task

To configure SSL to be FIPS-compliant, perform the following procedures:

- [Configuring FIPS compliance on page 336](#)
- Configure the Tivoli event integration facility port. See [Configuring the Tivoli event integration facility port on page 338](#).



Note:

If you are using dynamic workload broker for dynamic scheduling in your network, note that the workstation of type BROKER does not support SSL. All IBM Workload Scheduler workstations must communicate with the workstation of type BROKER using TCP/IP protocol.

Configuring FIPS compliance

Configuring FIPS compliance for your network.

About this task

Perform the following configuration steps to prepare the master domain manager and the Dynamic Workload Console for FIPS compliance.

- On both the master domain manager and the Dynamic Workload Console workstations, perform the following steps:
 - Configure IBM® JDK with FIPS enabled on the server. Create a backup and replace `JavaExt/jre` with `IBM_JDK_PATH>/jre`.

- b. Configure batch reports for FIPS. Edit the SDK `java.security` file in the path

`<IBM_JDK_PATH>/jre/lib/security/java.security` to insert the **IBMJCEFIPS** provider

(**com.ibm.crypto.fips.provider.IBMJCEFIPS**). **IBMJCEFIPS** must precede the **IBMJCE** provider in the provider list.

- i. In the `security.provider` list, modify the entry containing **IBMJCE** and add it to the top of the list as follows:

```
#
# List of providers and their preference orders (see above):
#
security.provider.1=com.ibm.crypto.fips.provider.IBMJCEFIPS
security.provider.2=com.ibm.jsse2.IBMJSSEProvider2
security.provider.3=com.ibm.crypto.provider.IBMJCE
security.provider.4=com.ibm.security.jgss.IBMJGSSProvider
security.provider.5=com.ibm.security.cert.IBMCertPath
security.provider.6=com.ibm.security.sasl.IBMSASL
security.provider.7=com.ibm.xml.crypto.IBMXMLCryptoProvider
security.provider.8=com.ibm.xml.enc.IBMXMLEncProvider
security.provider.9=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
security.provider.10=sun.security.provider.Sun
security.provider.11=com.ibm.security.cmskeystore.CMSProvider
```

- ii. On RedHat Enterprise Linux® server, check the **securerandom.source** property in the `java.security` file and ensure the value is specified as follows:

```
securerandom.source=file:/dev/./urandom
```

- c. Configure the WebSphere Application Server Liberty Base `jvm.options` file, located in `<TWA_DATA_DIR>/usr/servers/engineServer/configDropins/overrides/jvm.options` on the master, and in `<DWC_DATA_dir>/usr/servers/dwcServer/configDropins/overrides/jvm.options` on the Dynamic Workload Console, to enable FIPS as follows:

```
Dcom.ibm.jsse2.usefipsprovider=true
```

2. On the master domain manager workstation, perform the following steps:

- a. Comment the following properties in the `eif.templ` file located in the path: `<TWA_DATA_DIR>/stdlist/appserver/engineServer/temp/TWS/EIFListener/eif.templ` as follows:

```
#SSL_ChannelSSLTruststoreAlgorithm=SunX509
#SSL_ChannelSSLKeystoreAlgorithm=SunX509
```

- b. To prepare your environment for FIPS, set the following local options in the `localopts` file on every IBM® Workload Scheduler agent in the network:

```
SSL Fips enabled      = yes
nm SSL port          = 31113
```

```
SSL keystore file      = "<TWA_home>/TWS/ssl/GSKit/TWSClientKeyStore.kdb"
SSL certificate keystore label = "client"
SSL keystore pwd       = "<TWA_home>/TWS/ssl/GSKit/TWSClientKeyStore.sth"
```

Set the following local options for the CLI:

```
CLI SSL keystore file           = "<TWA_home>/TWS/ssl/GSKit/TWSClientKeyStore.kdb"
CLI SSL certificate keystore label = "client"
CLI SSL keystore pwd           = "<TWA_home>/TWS/ssl/GSKit/TWSClientKeyStore.sth"
```

where `<TWA_home>` is the installation directory of the instance of IBM® Workload Scheduler where the agent is installed.



Note: On Windows™ workstations, the user, **SYSTEM**, must have read-permissions to read the GSKit FIPS certificates.

For more information about setting local options and the `localopts` file, see [Setting local options on page 51](#)

- Restart the server on both the master domain manager and the Dynamic Workload Console workstation.
- On the dynamic agent workstations, add the following property to the `JVMOptions` in the `JobManager.ini` file:

```
-Dhttps.protocols=TLSv1.2
```

The `JobManager.ini` is located in:

On UNIX™ operating systems

```
<TWA_DATA_DIR>/ITA/cpa/config/JobManager.ini
```

On Windows™ operating systems

```
<TWA_home>\TWS\ITA\cpa\config\JobManager.ini
```

- Restart the agent workstation.

Configuring the Tivoli event integration facility port

About this task

The Tivoli event integration facility port for SSL, `eventProcessorEIFSSLPort`, is used in event management. For the Tivoli event integration facility port to communicate in FIPS mode, you must first configure WebSphere Application Server Liberty Base for FIPS. See the step related to configuring `jvm.options` in [Configuring FIPS compliance on page 336](#).

To configure the Tivoli event integration facility port for SSL, perform the following steps:

- Ensure that you have set the **SSL Fips Enabled** local option on every agent, as described in [Configuring FIPS compliance on page 336](#).
- Set the global option for the port by using `optman`, as follows:

```
eventProcessorEIFSSLPort / ef = portnumber
```

where *portnumber* is the number of any free port on your network.

- To update the Symphony file, run `JnextPlan -for 0000`.

4. Restart the EventProcessor by using the conman stopvtp and conman startvtp commands.
5. Restart the IBM Workload Scheduler monitoring engine with the conman commands, stopmon and startmon.

Finding the GSKit version on agents running on UNIX™ and Linux™ operating systems

About this task

To find which version of GSKit runs on your agent, go to following path depending on the version of GSKit and submit the appropriate command:

GSKit 32 bit

Path

```
/usr/Tivoli/TWS/GSKit32/8/bin
```

Command

```
gsk8ver
```

GSKit 64 bit

Path

```
/usr/Tivoli/TWS/GSKit64/8/bin
```

Command

```
gsk8ver_64
```

On UNIX™ and Linux™, you can optionally run the `ita_props.sh` script to set the environment to `/usr/Tivoli/TWS/GSKit32/8/bin` or `/usr/Tivoli/TWS/GSKit64/8/bin`, so that you can run this command directly without having to specify the relative path.

Chapter 7. Data maintenance

This chapter describes how to maintain your IBM Workload Scheduler database and other data files. The database is hosted on either the DB2® or Oracle RDBMS infrastructure, as you determined when you installed it. You should use the documentation of DB2® or Oracle for general instructions on database maintenance. This chapter describes the maintenance activities that are specific to IBM Workload Scheduler.

It comprises the following sections:

- [Maintaining the database on page 340](#)
- [Maintaining the file system on page 343](#)
- [Administrative tasks - DB2 on page 356](#)
- [Administrative tasks - Oracle on page 362](#)
- [Modifying your RDBMS server on page 363](#)
- [Keeping track of database changes using audit reports on page 383](#)
- [Collecting job metrics on page 388](#)

Maintaining the database

This section discusses the following:

- Backing up and restoring files in the IBM Workload Scheduler databases. See [Backing up and restoring on page 340](#).
- Ensuring that a backup master domain manager is as up-to-date as possible. See [Using a backup master domain manager with a backup database on page 340](#).
- Maintaining the performance level of the IBM Workload Scheduler databases. See [Reorganizing the database on page 342](#).

Backing up and restoring

To minimize downtime during disaster recovery, back up your master data files frequently to either offline storage or a backup master domain manager.

Using a backup master domain manager with a backup database

Set up a backup master domain manager that accesses a different database than the master domain manager, and get your database administrator to set up a mirror of the master domain manager's database onto the backup master domain manager's database. In this way your backup master domain manager not only receives copies of all the processing messages, as is provided for by the setting of the *FullStatus* attribute on the backup master domain manager, but is also able to access the mirrored database. The mirror frequency must be set high enough to match the frequency with which you change the database.

For more information about how to use a backup master domain manager, see [Switching the master to a backup on page 396](#).

Backing up the configuration files

The configuration files used by IBM Workload Scheduler are found in the following places:

useropts

<user_home_dir>/TWS

For the localopts, sfinal, Security and *.msg files

On Windows operating systems

<TWA_home>

On UNIX operating systems

<TWA_DATA_DIR>

IBM® Workload Scheduler configuration files

On Windows operating systems

<TWA_home>\TWS\mozart*.*

On UNIX operating systems

<TWA_DATA_DIR>/TWS/mozart/*.*

This directory might contain the following files:

runmsgno

This is used for the allocation of unique prompt numbers. On the master domain manager, this file should not be edited manually. On other workstations it can be edited only in the circumstances described in the *Troubleshooting Guide*. This file does not need to be backed up.

globalopts

This is used to store a copy of three of the global properties stored in the database. It must be edited only in the circumstances described in [Switching the master to a backup on page 396](#). This file should be backed up if it is edited.

Application server configuration files, such as TWSConfig.properties

On Windows operating systems

<TWA_home>\usr\servers\engineServer\Resources\properties

On UNIX operating systems

<TWA_DATA_DIR>/usr/servers/engineServer/Resources/properties

Forecast plan files

On Windows operating systems

<TWA_home>\TWS\schedForecast

On UNIX operating systems

<TWA_DATA_DIR>/TWS/schedForecast

Archived plan files.**On Windows operating systems**

```
<TWA_home>\TWS\schedlog
```

On UNIX operating systems

```
<TWA_DATA_DIR>/TWS/schedlog
```

Trial plan files**On Windows operating systems**

```
TWA_home\TWS\schedTrial
```

On UNIX operating systems

```
<TWA_DATA_DIR>/TWS/schedTrial
```

A detailed list of all files is not supplied, as there are too many files. Back up all the files in these directories.



Note: The `tw_inst_pull_info` tool (described in the *Troubleshooting Guide*) is provided for sending information to support, but can also be used to perform a backup of a DB2® database and some of the configuration files.

Backing up log files

Make a regular offline backup of all log files, identifying them from the information given in the section on log and trace files in the *IBM Workload Scheduler: Troubleshooting Guide*.

If you use `tw_inst_pull_info` for backup (see the documentation in the same guide), you do not need to separately backup these files.

Reorganizing the database

The database requires routine maintenance, as follows:

DB2®

The DB2® database has been set up to maintain itself, so there is little user maintenance to do. Periodically, DB2® checks the database by running an internal routine. DB2® determines when this routine must be run using a default policy. This policy can be modified, if need be, or can be switched off so that DB2® does not perform internal automatic maintenance. Using the statistical information that DB2® discovers by running this routine, it adjusts its internal processing parameters to maximize its performance.

This routine has also been made available for you to run manually in the case either where you feel that the performance of DB2® has degraded, or because you have just added a large amount of data, and anticipate performance problems. The routine is imbedded in a tool called `dbrunstats`, which can be run to improve performance while DB2® is processing data without causing any interruption.

It is also possible to physically and logically reorganize the database using the `dbreorg` script. This effectively re-creates the *tablespace* using its internal algorithms to determine the best way to physically and logically

organize the tables and indexes on disk. This process is time-consuming, and requires that IBM Workload Scheduler is down while it is run, but it does provide you with a freshly reorganized database after major changes.

The use of these tools is described in [Administrative tasks - DB2 on page 356](#).

These tools are implementations of standard DB2 facilities. If you are an expert user of DB2 you can use the standard facilities of DB2® to achieve the same results. For details go to the Information Center for DB2®, version 9.5, at: <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5//index.jsp>.

Oracle

For Oracle databases see the Oracle maintenance documentation.

Oracle 10g by default has an internally scheduled procedure to collect database statistics: if the default schedule is not changed, Oracle 10g will automatically optimize its performance by running this procedure daily. Oracle 9i does not have the same schedule by default, but could be set up to do so.

Maintaining the file system

Some of the file systems and directories need periodic maintenance. The details are given under the following topics:

- [Avoiding full file systems on page 343](#)
- [Log files and archived files on page 349](#)
- [Temporary files on page 355](#)
- [Managing event message queue file sizes on page 355](#)

Avoiding full file systems

Perhaps the most important maintenance task to perform is that of regularly controlling the file system or systems where IBM Workload Scheduler is installed, particularly on the master domain manager.

IBM Workload Scheduler has a number of files that can grow in size, either with more extensive use, such as the Symphony file, or in the event of network problems, such as the message files. If the Symphony file, in particular, cannot be expanded to contain all the required records, it might become corrupted. If this happens on a fault-tolerant agent or on a domain manager other than the master domain manager, there is a recovery procedure (see the *IBM Workload Scheduler: Troubleshooting Guide*). If the Symphony file on the master domain manager is corrupted, you have no alternative but to restart IBM Workload Scheduler, losing the current plan's workload.

It is thus *most important* that you monitor the available space on the file system of the master domain manager where the Symphony file is generated, to ensure that there is always sufficient space for it to expand to cover any workload peaks, and also that there is sufficient space for message files to expand in the event of network problems. Your experience with your workload and your network will guide you to determine what are the acceptable limits of available disk space.

The approximate size of the Symphony file can be estimated in advance. It contains items related both to the plan (see [Table 67: Algorithm for calculating the approximate size of the plan data in the Symphony file on page 344](#)) and to the database (see [Table 68: Algorithm for calculating the approximate size of the database data in the Symphony file on page 344](#)).

Estimate how many items you have in each category, multiply them by the indicated size in bytes, and sum them to find the approximate Symphony file size:

Table 67. Algorithm for calculating the approximate size of the plan data in the Symphony file

Data in Symphony file from the current plan	Bytes per instance
Per Job Scheduler instance:	512
Per job instance:	512
Per job "docommand" string > 40 bytes:	The length of the "docommand" string
Per ad hoc prompt:	512
Per file dependency:	512
Per recovery prompt:	512
Per recovery job:	512

Table 68. Algorithm for calculating the approximate size of the database data in the Symphony file

Data in Symphony file from the database (on the master domain manager)	Bytes per instance
Per workstation:	512
Per resource:	512
Per user:	256
Per prompt:	512
If the global option <code>ignoreCalendars</code> is set to <i>off</i> , per calendar:	512

If you find that disk space is becoming too limited, and you cannot dynamically extend it, you must create a backup master domain manager with much more space on its file system and then use the `switchmgr` command so that the backup becomes your new domain manager. Instructions on how to do this for any domain manager are given in [Switching a domain manager on page 391](#), and in particular for a master domain manager, in [Switching the master to a backup on page 396](#).

Monitoring the disk space used by IBM Workload Scheduler

You can use event-driven workload automation (EDWA) to monitor the disk space used by IBM Workload Scheduler and to start a predefined set of actions when one or more specific events take place. This type of event is managed by the `TWSApplicationMonitor` event provider. These types of events are supported on fault-tolerant agents only and not supported on dynamic agents. You can use EDWA to set up an event rule that monitors the used disk space, to verify that there is enough space to generate the Symphony and log files, and to allow the product to work correctly. For more information about event-driven workload automation, see the section about event-driven workload automation in the *User's Guide and Reference*.

When calculating disk space usage, the SSM agent divides the used space by the total space and then rounds up the result to the next highest integer. This is important to note especially when the usage is close to the specified threshold. See [Table](#)

69: Example for the ge operator on page 346 and Table 70: Example for the le operator on page 348 for examples on how this impacts the disk space usage calculation.

The following .XML file contains the definition of a sample event rule to monitor the disk usage percentage. This event rule triggers the MessageLogger action provider to write a message in a log file in an internal auditing database when the event occurs. For more information about the MessageLogger action provider, see IBM Workload Scheduler User's Guide and Reference :

```
<?xml version="1.0"?>
<eventRuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules"
  xsi:schemaLocation="http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules
  http://www.ibm.com/xmlns/prod/tws/1.0/event-management/rules/EventRules.xsd">
  <eventRule name="FILESYSTEMFULL" ruleType="filter" isDraft="yes">
    <eventCondition name="twsDiskMonEvt1" eventProvider="TWSApplicationMonitor" eventType="TWSDiskMonitor">
      <scope>
        * Disk is filling up
      </scope>
      <filteringPredicate>
        <attributeFilter name="FillingPercentage" operator="ge">
          <value>usage_percentage</value>
        </attributeFilter>
        <attributeFilter name="Workstation" operator="eq">
          <value>workstation_name</value>
        </attributeFilter>
        <attributeFilter name="SampleInterval" operator="eq">
          <value>sample_interval</value>
        </attributeFilter>
        <attributeFilter name="MountPoint" operator="eq">
          <value>mount_point</value>
        </attributeFilter>
      </filteringPredicate>
    </eventCondition>
    <action actionProvider="MessageLogger" actionType="MSGLOG" responseType="onDetection">
      <scope>
        OBJECT=ADWDAD MESSAGE=Disk is filling up
      </scope>
      <parameter name="ObjectKey">
        <value>object_key</value>
      </parameter>
      <parameter name="Severity">
        <value>message_severity</value>
      </parameter>
      <parameter name="Message">
        <value>log_message</value>
      </parameter>
    </action>
  </eventRule>
</eventRuleSet>
```

where:

usage_percentage

Is the disk usage percentage.

! **Important:** When creating the event rule, always use a whole integer in the range 1-99 (inclusive) to express the threshold value. Fractions, decimals and negative numbers are not supported and the event rule is ignored.

Supported operators are as follows:

ge

causes the event generation when the disk usage exceeds the percentage specified by the threshold value. If the condition described in the rule already exists when you deploy the rule, the related event is generated. If the condition does not exist at the time the rule is deployed, then the event is generated when the disk usage percentage reaches or exceeds the threshold. The event is generated again only if the disk usage percentage subsequently falls below the threshold value and then rises again and either reaches or exceeds the threshold. If you restart the SSM agent and the disk usage percentage is higher than the threshold value, the event is generated again.

[Table 69: Example for the ge operator on page 346](#) provides an example in which the **ge** operator is set to 70%.

Table 69. Example for the ge operator

Ma il box n ame	Disk usage percentage	A ct ion
Sa m ple (0)	>= 70%	ev ent ge ne ra ted
Sa m ple (0)	< 70%	ev ent not ge ne ra ted
Sa m	< 70%	ev ent

Table 69. Example for the ge operator (continued)

Ma il box n ame	Disk usage percentage	A ct ion
ple (n -1)		not ge ne ra ted
Sa >= 70% m ple (n)		ev ent ge ne ra ted
Sa >= 70% m ple (n +1)		ev ent not ge ne ra ted

le

causes the event generation when the disk usage percentage decreases under the threshold value. If the condition described in the rule already exists when you deploy the rule, the related event is not generated. The event is generated only the first time the specified disk usage percentage is reached. The event is generated again only if the disk usage percentage subsequently rises and exceeds the threshold value and then falls below the threshold. If you restart the SSM agent and the disk usage percentage is lower than the threshold value, the event is not generated until the disk usage percentage exceeds the threshold value and then falls below it again. [Table 70: Example for the le operator on page 348](#) provides an example in which the **le** operator is set to 50%:

Table 70. Example for the le operator

Ma il box name	Disk usage percentage	A ct ion
Sa m ple (0)	$\leq 50\%$	ev ent not ge ne ra ted
Sa m ple (0)	$> 50\%$	ev ent not ge ne ra ted
Sa m ple (n -1)	$> 50\%$	ev ent not ge ne ra ted
Sa m ple (n)	$\leq 50\%$	ev ent ge ne ra ted
Sa m ple (n +1)	$\leq 50\%$	ev ent not ge ne

Table 70. Example for the le operator (continued)

Ma il box n ame	Disk usage percentage	A ct ion
		ra ted

workstation_name

Is the workstation on which the event is generated.

sample_interval

Is the interval, expressed in seconds, for monitoring the disk usage percentage.

mount_point

Is the mount point of the file system where IBM Workload Scheduler is installed, for example: "C:" on Windows™ systems or "/" on UNIX™ systems.

object_key

Is a key identifying the object to which the message pertains.

message_severity

Is the severity of the message.

log_message

Is the message to be logged.

Log files and archived files

Log files are produced from a variety of IBM Workload Scheduler activities. Other activities produce files which are archived after they have been used. The details are given in [Table 71: Log and trace file maintenance on page 350](#).

When generating a job log in the monitoring section of the Dynamic Workload Console, by default the master domain manager generates a temporary file in the `/tmp` folder for the job log for each submitted job.

You can optionally avoid the temporary download of the job log on the master domain manager if the job log contains confidential information. This ensures compliance with the PCI standard.

Follow these steps to change this behavior:

1. Set the **com.ibm.tws.conn.plan.output.logtype** property in the `TWSConfig.properties` file to `memory`
2. Stop and start all the IBM® Workload Scheduler processes

The temporary job log file will be loaded into memory, not leaving any track within the server system and console file. This will result in the product returning to PCI requirements. If no request arrives within the timeout specified in the **ccom.ibm.tws.conn.plan.output.timeout**, the operation is canceled.

Table 71. Log and trace file maintenance

Activity	Description	Location	Maintenance method
Fault-tolerant agent	Each IBM Workload Scheduler process logs its activities, writing them in log and trace message files:	netman	rmstdlist
	<p>Log messages</p> <p>These are messages intended for use directly by you, and provide information, errors and warnings about the processes.</p>		

Trace messages

Table 71. Log and trace file maintenance (continued)

Activity	Description	Location	Maintenance method
	These are messages written when a problem occurs that you can probably not solve without the assistance of IBM Software Support.	<p>netman</p> <p>On Windows systems</p> <p><i>TWA_home</i> /TWS/st dlist/traces/yy yyymmdd_NETMAN. log</p> <p>On UNIX operating systems</p> <p><<i>TWA_DATA_DIR</i>>/ stdlist/logs/yy yyymmdd_NETMAN. log</p> <p>Other processes</p> <p>On Windows systems</p> <p><i>TWA_home</i> /TWS/st dlist/traces/yy yyymmdd_TWSMERGE .log</p> <p>On UNIX operating systems</p> <p><<i>TWA_DATA_DIR</i>>/ stdlist/logs/yy yyymmdd_TWSMERGE .log</p>	
Master domain manager job management	The job manager process on the master domain manager archives the previous period's Symphony file.	<p>This is the default situation. You can set an option in the <code>localopts</code> file to create separate trace files for the major processes.</p> <p>On Windows systems</p> <p><i>TWA_home</i>>\TWS\schedlog\date</p> <p>On UNIX systems</p> <p><<i>TWA_DATA_DIR</i>>/TWS/schedlog/date</p>	Manual

Table 71. Log and trace file maintenance (continued)

Activity	Description	Location	Maintenance method
Job	Each job that runs under IBM Workload Scheduler control creates an output file. These files are archived.	<p>On Windows systems</p> <p><i>TWA_home</i>\TWS\stdlist\<i>date</i></p> <p>On UNIX systems</p> <p><<i>TWA_DATA_DIR</i>>/TWS/stdlist/<i>date</i></p> <p>where <i>date</i> is in the format <i>yyyy.mm.dd</i></p>	rmstdlist
Dynamic agent	Log messages	<p>Windows</p> <p><i>TWA_home</i>>\TWS\stdlist\JM\JobManager_message.log</p> <p>UNIX</p> <p><<i>TWA_DATA_DIR</i>>/TWS/stdlist/JM/JobManager_message.log</p>	Regular housekeeping is performed through the configuration of several parameters. See Regular maintenance on page 95 .
	Trace messages	<p>Windows</p> <ul style="list-style-type: none"> • <i>TWA_home</i>>\TWS\stdlist\JM\ITA_trace.log • <i>TWA_home</i>>\TWS\stdlist\JM\JobManager_trace.log • <i>TWA_home</i>>\TWS\JavaExt\logs\javaExecutor0.log 	

Table 71. Log and trace file maintenance (continued)

Activity	Description	Location	Maintenance method	
Jobs with advanced options	UNIX	<ul style="list-style-type: none"> • <TWA_DATA_DIR>/TWS/stdlist/JM/ITA_trace.log • <TWA_DATA_DIR>/TWS/stdlist/JM/JobManager_trace.log • <TWA_DATA_DIR>/TWS/JavaExt/logs/javaExecutor0.log 		
		Windows		<pre>TWA_home>\TWS\stdlist\JM \date></pre>
		UNIX		<pre><TWA_DATA_DIR>/TWS/stdlist/JM/date></pre>
		where <i>date</i> is in the format <i>yyyy.mm.dd</i>		
Forecast and trial plan creation	The creation of forecast and trial plans require manual maintenance.	<p>Forecast plan</p> <p>These files are to be maintained manually</p> <p>Trail plan</p> <p>These files are to be maintained manually</p>	Manual	
Audit	The audit facility writes log files.	<p>On Windows operating systems</p> <pre>TWA_home/TWS/audit</pre> <p>On UNIX operating systems</p> <pre><TWA_DATA_DIR>/audit</pre>	Manual	
DB2® UDB	DB2® logs its activities.	Information about the location and viewing method for DB2® log files is supplied in the	See the DB2® documentation.	

Table 71. Log and trace file maintenance (continued)

Activity	Description	Location	Maintenance method
		DB2® documentation, in the Knowledge Center for DB2®.	
		The main file to control is the <code>db2diag.log</code> file, which is the most important DB2® diagnostic file, which, without intervention, grows endlessly with no reuse of wasted space. This does not apply, however, to the database log files used by IBM Workload Automation, which are set up for circular reuse of disk space, so they don't grow in size over a maximum value.	
Oracle database	Oracle logs its activities.	See the Oracle documentation.	See the Oracle documentation.
WebSphere Application Server Liberty Base	The application server writes log files.	<p>On the master components:</p> <p>On Windows operating systems</p> <pre>TWA_home>\TWS\stdlist\applicationserver\engineServer\logs</pre> <p>On UNIX operating systems</p> <pre><TWA_DATA_DIR>/stdlist/applicationserver/engineServer/logs</pre> <p>On the Dynamic Workload Console:</p> <p>On Windows operating systems</p> <pre>TWA_home>\stdlist\applicationserver\dwcServer\logs</pre> <p>On UNIX operating systems</p> <pre><TWA_DATA_DIR>/stdlist/applicationserver/dwcServer/logs</pre>	Manual
Netcool® SSM monitoring agent (not supported on IBM i systems)	The agent writes log files. (<code>ssmagent.log</code> , <code>traps.log</code>)	<p>On Windows operating systems</p> <pre>TWA_home>\TWS\ssm\Log</pre> <p>On UNIX operating systems</p> <pre><TWA_DATA_DIR>/EDWA/ssm/Log/</pre>	Manual

Table 71. Log and trace file maintenance (continued)

Activity	Description	Location	Maintenance method
Other	Other activities also write trace and log files.	<p>On Windows operating systems</p> <p><i>TWA_home>\TWS\methodes</i></p> <p>On UNIX operating systems</p> <p><i>TWA_home>/TWS/methods</i></p>	Manual

The easiest method of controlling the growth of these directories is to decide how long the log files are needed, then schedule a IBM Workload Scheduler job to remove any files older than the given number of days. Use the `rmstdlist` command for the process and job log files, and use a manual date check and deletion routine for the others. Make sure that no processes are using these files when you perform these activities.

See the *User's Guide and Reference* for full details of the `rmstdlist` command.



Note: The `rmstdlist` command might give different results on different platforms for the same scenario. This is because on UNIX® platforms the command uses the `-mtime` option of the `find` command, which is interpreted differently on different UNIX® platforms.

Temporary files

The IBM Workload Scheduler master domain manager uses temporary files, located in `<TWA_home>/TWS/tmp` or `/tmp` and named `TWS<XXXX>`, when compiling new production control databases. These files are deleted when compiling is complete.

This directory also contains the IBM Workload Scheduler installation files and log files, it is primarily used to handle temporary files that composer CLI creates as a work repository when it is invoked to perform CRUD actions against the IBM Workload Scheduler modeling objects. Directory rights are set to 777 to allow all users running the composer to have access. For security reasons the composer CLI is defined by using the sticky bit, so the files it creates can be owned by users different from the IBM Workload Scheduler installation user. IBM Workload Scheduler `conman` can be used by any user therefore the folder is 777. If the users eligible to use `conman/composer` are inserted into the IBM Workload Scheduler group then the permission can be set to 774. In that way, only these users will be able to run `conman/composer` commands

Managing event message queue file sizes

This publication contains the following information with respect to managing event message queue file sizes:

- See [Planning space for queues on page 280](#) to learn about planning space for message event queues (and also how to use `evtsize` to resize the queues)
- See [Managing the event processor on page 426](#) to learn about managing the EIF event queue

- See [Disk Space on page 471](#) to learn about the impacts that increased fault tolerance can have on message queues
- See [Workload spreading on page 468](#) to learn about how to avoid bottlenecks in the Mailbox.msg queue.

Administrative tasks - Databases

A set of scripts and SQL files is provided for each database type to perform actions such as granting rights or reorganizing the database. These files are located in `inst_dir/TWS/dbtools` into a separate folder for each database type. To use these files, copy the relevant folder to the database server. The available files are as follows:

DB2

The DB2® tools must be run by a user who has the following permissions:

- DB2 administrator permissions – the user must be defined to DB2 as a DB2 Administrator
- Full access (777) to the IBM® Workload Scheduler installation directory

dbrunstats

This script runs the DB2 statistics program, to maximize the performance of DB2.

dbreorg

This script reorganizes the database. See [Reorganizing the DB2 database on page 359](#) for a full description of how to use the tool.

dbgrant

This script adds grants to new users on IBM® Workload Scheduler DB schema for the views that can be used to generate reports

Oracle, Informix, MSSQL

dbgrant

This script grants the user permissions for the Dynamic Workload Console views. See the Dynamic Workload Console online help for full details.



Note: The tools in this directory might include one that is only for the use of IBM Software Support:

dbmove

Do not run this script. To do so might damage or overwrite the data in your database.

Administrative tasks - DB2®

This section describes how to perform some specific administrative tasks on DB2®, as follows:

- [Changing DB2 passwords on page 357](#)
- [Locating the DB2 tools on page 357](#)
- [User permissions for running the DB2 tools on page 357](#)

- [Running DB2 maintenance manually on page 358](#)
- [Reorganizing the DB2 database on page 359](#)
- [Monitoring the lock list memory on page 360](#)

Changing DB2® passwords

About this task

To change passwords used by DB2® other than the `<TWS_user>` password or the passwords of the user IDs used by IBM Workload Scheduler to access the database (see [Changing key IBM Workload Scheduler passwords on page 412](#)) follow the instructions in the DB2® documentation.

After you have changed the password, modify the **db.password** parameter in the `datasource_db2.xml` configuration file, as described in [Changing the properties for the database on page 418](#). You can optionally encrypt the password, as described in the topic about encrypting passwords in *IBM Workload Scheduler: Planning and Installation*.

Locating the DB2® tools

About this task

IBM Workload Scheduler is supplied with a small set of tools that you use to perform the following administrative tasks for DB2®:

- Run the DB2® statistics program, to maximize the performance of DB2® (`dbrunstats`). See [Running DB2 maintenance manually on page 358](#) for a full description of how to use the tool.
- Reorganize the database (`dbreorg`). See [Reorganizing the DB2 database on page 359](#) for a full description of how to use the tool.

The tools are available in the following directory:

```
inst_dir/TWS/dbtools/db2
```

Copy the tools to the DB2 server workstation where the IBM Workload Scheduler database is located.



Note: The tools in this directory might include some that are for the use of IBM Software Support:

`dbmove`

Do not run this script. To do so might damage or overwrite the data in your database.

User permissions for running the DB2® tools

The DB2® tools must be run by a user who has the following permissions:

- DB2® administrator permissions – the user must be defined to DB2® as a DB2® Administrator
- Full access (777) to the IBM Workload Scheduler installation directory

Running DB2® maintenance manually

At installation, DB2® automatic maintenance is switched on, which means that DB2® periodically checks to see if it needs to collect new database statistics, so that it can perform the maintenance, adjusting the performance parameters to maximize performance.

This section describes how to perform the DB2® maintenance process on demand, instead of waiting for DB2® to do it according to its automatic maintenance policy. The process is run by the `dbrunstats` tool which you can run whenever you need to, without stopping DB2® or interrupting its processing.

To run this tool, follow this procedure:

1. Locate the DB2® tools: see [Locating the DB2 tools on page 357](#).
2. Check that the user who is going to run the procedure has the appropriate rights (see [User permissions for running the DB2 tools on page 357](#))
3. On the DB2 server, open a DB2® shell, as follows:

UNIX™

Follow these steps:

- a. Issue the command `su - db2inst1`, or change to the subdirectory `sqllib` of the home directory of the owner of the DB2® instance (by default `db2inst1`)
- b. Launch the command `./db2profile`

Windows™

Select from the **Start** menu, **Programs** →; **IBM DB2** →; **Command Line Tools** →; **Command Window**

4. Check that the command shell is correctly initialized by issuing the command `db2`, and checking that the command is recognized.
5. Issue the command `quit` to leave the DB2® Processor mode.
6. From within the shell, browse to the directory where you copied the script.
7. Run the script:

UNIX™

```
dbrunstats.sh database [user [password]]
```

Windows™

```
dbrunstats database [user [password]]
```

where:

database

The name of the database. The default name is `TWS`. Supply this value unless you have changed it.

user

The DB2® administration user. If this is omitted, the ID of the user running the command will be used.

password

The password of the DB2® administration user. If this is omitted, it will be requested interactively.

The script runs, giving you various messages denoting its progress and successful conclusion. At the end (it is not particularly time-consuming) the database performance parameters have been reset to maximize performance.

Reorganizing the DB2® database

About this task

Using this tool, the database physically reorganizes the data tables and indexes, optimizing disk space usage and ease of data access. The process is time-consuming, requires that the database is backed up, and that IBM Workload Scheduler is stopped. However, at the end you have a database that is completely reorganized.

To reorganize the database follow this procedure:

1. Stop WebSphere Application Server Liberty Base and appservman by running the following command:

```
conman "stopappserver;wait"
```

See [Starting and stopping the application server and appservman on page 436](#) for full details.

2. Back up the IBM Workload Scheduler database. Follow the instructions in the database vendor documentation, as appropriate.
3. Check that the user who is going to run the procedure has the appropriate rights (see [User permissions for running the DB2 tools on page 357](#))
4. On the DB2 server where you copied the script, open a DB2® shell, as follows:

UNIX™

Follow these steps:

- a. Issue the command `su - db2inst1`, or change to the subdirectory `sqllib` of the home directory of the owner of the DB2® instance (by default `db2inst1`)
- b. Launch the command `./db2profile`

Windows™

Select from the **Start** menu, **Programs** →; **IBM DB2** →; **Command Line Tools** →; **Command Window**

5. Check that the command shell is correctly initialized by issuing the command `db2`, and checking that the command is recognized.
6. Issue the command `quit` to leave the DB2® Processor mode.
7. From within the shell, browse to the directory where you copied the script.
8. Run the script:

UNIX™

```
dbreorg.sh database [user [password]]
```

Windows™

```
dbreorg database [user [password]]
```

where:

database

The name of the database. The default name is `TWS`. Supply this value unless you have changed it.

user

The DB2® administration user. If this is omitted, the ID of the user running the command will be used.

password

The password of the DB2® administration user. If this is omitted, it will be requested interactively.

The script runs, giving you various messages denoting its progress and successful conclusion.

- Restart WebSphere Application Server Liberty Base and appservman by running the following command:

```
conman "startappserver;wait"
```

See [Starting and stopping the application server and appservman on page 436](#) for full details.

Monitoring the lock list memory

About this task

If the memory that DB2® allocates for its lock list begins to be fully used, DB2® can be forced into a "*lock escalation*", where it starts to lock whole tables instead of just individual table rows, and increasing the risk of getting into a deadlock.

This happens especially when there are long transactions, such as the creation or extension of a plan (production, trial, or forecast).

To avoid this problem occurring, set the automatic notification in the DB2® Health Center, so that you can be advised of any lock list problems building up.

However, if you think that deadlock situations have been occurring, follow this procedure to verify:

- With the WebSphere Application Server Liberty Base active, log on as DB2® administrator to the DB2® server, for example,

```
su - db2inst1
```

- Run the following command to determine where the IBM Workload Scheduler database is located:

```
db2 list active databases
```

The output might be as follows:

```
Database name           = TWS
Applications connected currently = 2
Database path           = /home/db2inst1/db2inst1/NODE0000/SQL00002/
```

- Run:

```
cd <Database_path>/db2event/db2detaildeadlock
```


4. Connect to the IBM Workload Scheduler database, for example:

```
db2 connect to TWS
```

5. Flush the event monitor that watches over deadlocks (active by default) with the following:

```
db2 flush event monitor db2detaildeadlock
```

6. Disconnect from the database with:

```
db2 terminate
```

7. Obtain the event monitor output with:

```
db2evmon -path . > deadlock.out
```

The file `deadlock.out` now contains the complete deadlock history since the previous flush operation.

8. To find out if there have been deadlocks and when they occurred, run:

```
grep "Deadlock detection time" deadlock.out
```

The output might be as follows:

```
Deadlock detection time: 11/07/2008 13:02:10.494600
Deadlock detection time: 11/07/2008 14:55:52.369623
```

9. But the fact that a deadlock occurred does not necessarily mean that the lock list memory is inadequate. For that you need to establish a relationship with lock escalation. To find out if there have been lock escalation incidents prior to deadlocks, run:

```
grep "Requesting lock as part of escalation: TRUE" deadlock.out
```

The output might be as follows:

```
Requesting lock as part of escalation: TRUE
Requesting lock as part of escalation: TRUE
```

If there has been lock escalation related to deadlocks, it is a good idea to modify the values of the following parameters.

LOCKLIST

This configures, in 4KB pages, the amount of memory allocated to locking management

MAXLOCKS

This configures the percentage of the memory that a single transaction can use, above which DB2® escalates, even though the memory might not be full

10. To determine the values currently being applied to the IBM Workload Scheduler database, do the following:

```
db2 get db cfg for TWS | grep LOCK
```

The output might be as follows:

Max storage for lock list (4KB)	(LOCKLIST) = 8192
Percent. of lock lists per application	(MAXLOCKS) = 60
Lock timeout (sec)	(LOCKTIMEOUT) = 180

The example shows the typical output for the IBM Workload Scheduler database if no modification has taken place to these values:

- "8192" = 4KB x 8192 pages = 32 MB of memory
- "60" = 60% – the percentage of memory that a single transaction can occupy before triggering an escalation
- "180" = 3 minutes of timeout for the period a transaction can wait to obtain a lock

11. The most straightforward action to take is to double the amount of memory to 64MB, which you do with the command:

```
db2 update db cfg for TWS using LOCKLIST 16384 immediate
```

12. Alternatively, you can set DB2® to automatically modify the LOCKLIST and MAXLOCKS parameters according to the amount of escalation being experienced and the available system memory. This self-tuning is a slow process, but adapts the database to the needs of the data and the available system configuration. It is done by setting the values of these parameters to AUTOMATIC, as follows:

```
db2 update db cfg for TWS using LOCKLIST AUTOMATIC immediate
```

DB2® responds with messages telling you that MAXLOCKS has also been set to AUTOMATIC:

```
SQL5146W "MAXLOCKS" must be set to "AUTOMATIC" when "LOCKLIST" is "AUTOMATIC".
```

```
"MAXLOCKS" has been set to "AUTOMATIC"
```



Note: The self-tuning facility is only available from V9.1 of DB2®.

Administrative tasks - Oracle

This section describes how to perform some specific administrative tasks for the Oracle database.

- [Changing the Oracle access password on page 362](#)
- [Maintaining the Oracle database on page 363](#)
- [Obtaining information about the IBM Workload Scheduler databases installed on an Oracle instance on page 363](#)
- [User permissions for running the Oracle tools on page 363](#)
- [Changing the properties for the database on page 418](#)

Changing the Oracle access password

About this task

This is described as part of the process of changing the password for a master domain manager or backup master domain manager. See [Changing key IBM Workload Scheduler passwords on page 412](#).

Maintaining the Oracle database

Like DB2, Oracle has a routine that regularly maintains the database. Similarly, this too can be run manually. The tool is invoked as follows:

```
dbms_stats.gather_schema_stats schema_owner
```

See the Oracle documentation for full details of how and when to run it.

Obtaining information about the IBM Workload Scheduler databases installed on an Oracle instance

About this task

To determine which IBM Workload Scheduler databases are installed on an Oracle instance, do the following:

```
su - oracle (UNIX only)
sqlplus system/system_password@service_name
SQL> select * from all_tws_schemas;
```

The output should look like the following:

```
SCHEMA_NAME
-----
MDL
mdm85<TWS_user>
```



Note:

1. More than one instance of IBM Workload Scheduler can be shared in one instance of Oracle, using different schemas.
2. In Oracle, the concept of "schema" and "user" are the same, so dropping an Oracle schema means dropping an Oracle user, which you do as follows:

```
SQL> drop user MDL cascade;
```

User permissions for running the Oracle tools

The Oracle tools must be run by a user who has the following permissions:

- Oracle administrator permissions – the user must be defined to Oracle as an administrator
- Full access (777) to the IBM Workload Scheduler installation directory

Modifying your RDBMS server

About this task

If you want to upgrade your database version, change the instance owner, or relocate it to a different host, complete the following steps:

1. If you are changing DB2®, check the *node directory* and *database directory* and make a note of the current configuration. To do this, issue the following commands at the DB2® command-line:

```
db2 list database directory
```

where the `show detail` attribute is specified to give the full information in the directory.

Make a note of the displayed details.

2. Stop the application server, using the command

```
conman stopappserver ;wait
```

3. Make the upgrade, instance owner change, or relocation, of the database following the instructions from your database supplier.
4. If you have changed the database host, port, or database name, you will need to update the application server's data source properties, as described in [Changing the properties for the database on page 418](#).
5. If you have changed the database access credentials, you will need to update the application server's security properties, as described in [Changing the security settings on page 424](#).
6. Reconfigure the database for IBM Workload Scheduler, as follows:

DB2®

- a. Check the *node directory* and *database directory*, as you did in [step 1 on page 364](#)
- b. If necessary, modify the data displayed by these commands to match the data you noted in [step 1 on page 364](#). If you are not certain of how to do this, contact IBM Software Support for assistance.

Oracle

Check the Oracle Listener and make sure that the service name is correctly specified.

7. Restart the database.
8. Restart the application server, using the command:

```
conman startappserver ;wait
```

Auditing facilities

Describes the audit facilities to track changes in the database and the plan, as well as those that track changes to objects involved in dynamic workload scheduling.

In the Dynamic Workload Console, operators and schedulers can review all changes to scheduling objects, both in the database and in the plan, discover which user performed a specific change, and when the change was performed. Administrators can request that each user provide a justification when making changes to an object and log this information in audit trails. Application developers and schedulers can compare and restore previous versions of each changed object, and promote the job workflow from development to test or production environments.

For more information, see the section about keeping track of changes in *Dynamic Workload Console User's Guide*.

Audit trails are useful to check enforcement and effectiveness of IT controls, for accountability, and vulnerability and risk analysis. IT organizations can also use auditing of security-related critical activities to aid in investigations of security incidents. When a security incident occurs, audit trails enable analysis of the history of activities (who did what, when, where, and how) that occurred prior to the security incident, so appropriate corrective actions can be taken. For these reasons, audit trails might need to be archived and accessible for years.

Two separate audit trail facilities are provided:

- Database and plan change tracking - see [Database and plan audit on page 365](#)
- Tracking of changes to scheduling objects to support dynamic workload scheduling - see [Dynamic workload scheduling audit on page 373](#)

Database and plan audit

An auditing option is available to track changes to the database and the plan. It is disabled by default. It is described in these sections:

- [Enabling and storing audit trails on page 365](#)
- [Audit log header format on page 367](#)
- [Audit log body format on page 368](#)
- [Sample audit log entries on page 372](#)

Enabling and storing audit trails

You can maintain audit trails for information stored in the database and in the plan. By default, auditing is enabled. To disable auditing, use the following global options:

enDbAudit

Enables auditing of the information available in the database.

enPlanAudit

Enables auditing of the information available in the plan.

For more information about global options, see [Global options - detailed description on page 29](#).

You can store auditing information in a file, in the IBM® Workload Scheduler database, or in both. To define in which type of store to log the audit records, use the **auditStore** global option. For more information about global options, see [Global options - detailed description on page 29](#). When auditing database information, all the user modifications are logged, including the current definition of each modified database object. If an object is opened and saved, the action is logged even if no modification was made. When auditing plan information, all the user modifications to the plan are logged. Actions are logged whether or not they are successful.

Choose the storage location of audit records according to the type of information you are auditing, whether it is database or plan:

auditing of the information available in the database (enDbAudit global option)

You can track changes to the database in a file, in the database itself, or in both. To define which type of store to log the audit records, use the **auditStore** global option. For more information about global options, see [Global options - detailed description on page 29](#). All the user modifications are logged, including the current definition of each modified database object. If an object is opened and saved, the action is logged even if no modification was made.

auditing of the information available in the plan (enPlanAudit global option)

You can track changes to the plan in a file. When you enable auditing of the information available in the plan, the information is saved to a file. All the user modifications to the plan are logged. Actions are logged whether or not they are successful.

Storing auditing information in a file (auditStore=file)

This storage location is available when you audit information in the database (**enDbAudit** global option) and in the plan (**enPlanAudit** global option). Choose to store auditing information in a file by setting the **auditStore** global option to `file`. For more information about the **auditStore** global option, see [Global options - detailed description on page 29](#).

Each audit log provides audit information for one day, from 00:00:00 UTC to 23:59:59 UTC regardless of the time zone of the local workstation, but the log file is created only when an action is performed or the WebSphere Application Server Liberty Base is started.

The files are called `yyyymmdd`, and are created in the following directories:

```
<TWA_home>/TWS/audit/plan <TWA_home>/TWS/audit/database
```

Audit entries are logged to a flat text file on individual workstations in the IBM Workload Scheduler network to minimize the risk of audit failure due to network issues. The log formats are the same for both plan and database. The logs consist of a header portion which is the same for all records, an action ID, and a section of data that varies according to the action type. All data is kept in clear text and formatted to be readable and editable from a text editor such as vi or notepad.

For more information about the details available in the logs, see [Audit log header format on page 367](#) and [Audit log body format on page 368](#).



Note: For modify commands, two entries are made in the log for resources, calendars, parameters, and prompts. The modify command is displayed in the log as a combination of the delete and add commands.

Storing auditing information in the database (auditStore=db)

This storage location is available when you audit information in the database (**enDbAudit** global option). Choose to store auditing information in the database by setting the **auditStore** global option to `db`. For more information about the **auditStore** global option, see [Global options - detailed description on page 29](#).

The AUDIT_STORE_RECORDS_V table is created in the IBM Workload Scheduler database.

For more information, see the section about the `AUDIT_STORE_RECORDS_V` table in *IBM Workload Scheduler: Database Views*.

Storing auditing information both in the database and in a file (`auditStore=both`)

This storage location is available when you audit information in the database (`enDbAudit` global option). Choose to store auditing information both in the database and in a file by setting the `auditStore` global option to `both`. For more information about the `auditStore` global option, see [Global options - detailed description on page 29](#).

For details about how the information is stored, see [Storing auditing information in the database \(`auditStore=db`\) on page 366](#) and [Storing auditing information in a file \(`auditStore=file`\) on page 366](#).

Audit log header format

Each log file starts with a header record that contains information about when the log was created and whether it is a plan or database log.

The header record fields are separated by vertical bars (`|`), as follows:

```
HEADER | <GMT_date> | <GMT_time> | <local_date> | <local_time> | <object_type> | >
      <workstation> | <user_ID> | <version> | <level>
```

Log Type

HEADER

GMT Date

The GMT date when the log file was created.

GMT Time

The GMT time when the log file was created.

Local Date

The local date when the log file was created. The local date is defined by the time zone option of the workstation.

Local Time

The local time when the log file was created. The local time is defined by the time zone option of the workstation.

Object Type

DATABASE for a database log file and PLAN for a plan log file.

Workstation Name

The IBM Workload Scheduler workstation name for which this file was created. Each workstation in the IBM Workload Scheduler network creates its own log.

User ID

The IBM Workload Scheduler user ID that created the log file.

Version

The version of the file.

Level

The logging level.

Audit log body format

The audit log formats are basically the same for the plan and the database. The log consists of a timestamp, a series of tags which identify the audit, object, action type, and data sections that vary with the action type. The data is in clear text format and each data item is separated by a comma (,).

The log file entries are in the following format:

```
"timestamp": "timestamp", "auditType": "audit_type",  
"objectType": "object_type", "actionType": "action_type",  
"workstationName": "workstation_name", "userName": "user_name",  
"frameworkUser": "framework_user", "objectName": "object_name"  
"actionDependentContents": "action-dependent_fields"
```

The log files contain the following information:

timestamp

Displays the date and time the action was performed in GMT time. The format is *yyyy-mm-dd:hh-mm-ss*.

auditType

Displays an eight-character value indicating the source of the log record. The following log types are supported:

CONMAN

conman command text

DATABASE

Database action

HEADER

The log file header

MAKESEC

makesec run

PARMS

Parameter command text

PLAN

Plan action

RELEASE

release command text

STAGEMAN

stageman run

objectType

Displays the type of the object that was affected by an action, from the following:

DATABASE

Database definition (for header only)

DBCAL

Database calendar definition

DBDOMAIN

Database domain definition

DBJBSTRM

Database Job Scheduler definition

DBJOB

Database job definition

DBPARM

Database parameter definition

DBPROMPT

Database prompt definition

DBRES

Database resource definition

DBSEC

Database security

DBUSER

Database user definition

DBVARTAB

Database variable table definition

DBWKCLS

Database workstation class definition

DBWKSTN

Database workstation definition

PLAN

Plan (for header only)

PLDOMAIN

Plan domain

PLFILE

Plan file

PLJBSTRM

Plan Job Scheduler

PLJOB

Plan job

PLPROMPT

Plan prompt

PLRES

Plan resource

PLWKSTN

Plan workstation

actionType

Displays what action was performed on the object. The appropriate values for this field are dependent on which action is being performed.

For the plan, the "*action_type*" can be ADD, DELETE, MODIFY, or INSTALL.

For the database, the ADD, GET, DELETE and MODIFY actions are recorded for workstation, workstation classes, domains, users, jobs, job streams, calendars, prompts, resources and parameters in the database.

The "**actionType**" field also records the installation of a new Security file. When **makesec** is run, IBM Workload Scheduler records it as an INSTALL action for a Security definition object.

LIST and DISPLAY actions for objects are not logged.

For parameters, the command line with its arguments is logged.

workstationName

Displays the IBM Workload Scheduler workstation from which the user is performing the action.

userName

Displays the logon user who performed the particular action. On Windows® operating systems, if the user who installed WebSphere® Liberty was a domain user, for Log Types **stageman** and **conman** this field contains the fully qualified user ID *domain\user*.

frameworkUser

Displays the framework user.

objectName

Displays the fully qualified name of the object. The format of this field depends on the object type as shown here:

DATABASE

N/A

DBCAL

"calendar"

DBDOMAIN

"domain"

DBJBSTRM

"workstation"#"job_stream"

DBJOB

"workstation"#"job"

DBPARM

"workstation"#"parameter"

DBPROMPT

"prompt"

DBRES

"workstation"#"resource"

DBSEC

N/A

DBUSER

["workstation"#]"user"

DBVARTAB

"variable_table"

DBWKCLS

"workstation_class"

DBWKSTN*"workstation"***PLAN**

N/A

PLDOMAIN*"domain"***PLFILE***"workstation"#"path"("qualifier")***PLJBSTRM***"workstation"#"job_stream_instance"***PLJOB***"workstation"#"job_stream_instance"."job"***PLPROMPT***["workstation"#]"prompt"***PLRES***"workstation"#"resource"***PLWKSTN***"workstation"***actionDependentContents**

Displays the action-dependent data fields. The format of this data is dependent on the *"actionType"* field.

Sample audit log entries

This is a sample database audit log:

```

HEADER |20080207|084124|20080207|094124|DATABASE|      |WK1|      | | |Version=A1.0| Level=1
DATABASE|20080207|084124|20080207|094124|DBRES  |ADD  |WK1|operator1|res=WK1#RESOURCE  |
DATABASE|20080207|100524|20080207|110524|DBWKSTN |MODIFY|WK1|operator1|ws=TIVOLI10      |
DATABASE|20080207|100525|20080207|110525|DBWKSTN |MODIFY|WK1|operator1|ws=ASLUTRI1      |
DATABASE|20080207|100525|20080207|110525|DBWKSTN |MODIFY|WK1|operator1|ws=WK1          |
DATABASE|20080207|100526|20080207|110526|DBDOMAIN|MODIFY|WK1|operator1|dom=MASTERDM      |
DATABASE|20080207|100610|20080207|110610|DBWKSTN |MODIFY|WK1|operator1|ws=TIVOLI10      |

```

```

DATABASE|20080207|100610|20080207|110610|DBWKSTN |MODIFY|WK1|operator1| |ws=ASLUTRI1 |
DATABASE|20080207|100611|20080207|110611|DBWKSTN |MODIFY|WK1|operator1| |ws=WK1 |
DATABASE|20080207|100611|20080207|110611|DBWKSTN |ADD |WK1|operator1| |ws=WK2 |
DATABASE|20080207|100612|20080207|110612|DBDOMAIN|MODIFY|WK1|operator1| |dom=MASTERDM |

```

This is a sample plan audit log:

```

HEADER |20080207|100758|20080207|110758|PLAN | |WK1|admin| | |Version=A1.0|Level=1
STAGEMAN|20080207|100758|20080207|110758|PLAN |INSTALL|WK1|admin| |C:\IBM\TWS\oper1\Symphony|
AWSBH030I The new Symphony file is installed.
STAGEMAN|20080207|100758|20080207|110758|PLAN |INSTALL|WK1|admin| |C:\IBM\TWS\oper1\Sinfonia|
AWSBH036I Multi-workstation Symphony file copied to C:\IBM\TWS\oper1\Sinfonia
STAGEMAN|20080207|100758|20080207|110758|ADITLEVL|MODIFY |WK1|admin| | |
AWSBH077I Audit level changing from 0 to 1.
CONMAN |20080207|100800|20080207|110800|PLWKSTN |MODIFY | |admin| |WK1 |
continue & start
CONMAN |20080207|100941|20080207|110941|PLWKSTN |MODIFY | |admin| |SLUTRI1 |
limit cpu=slutril;10
PLAN |20080207|101018|20080207|111018|PLWKSTN |MODIFY |WK1|oper1| |WK1 |
limit cpu=SLUTRI1;20
PLAN |20080207|101028|20080207|111028|PLDOMAIN|MODIFY |WK1|oper1| |ECCOLO |
reply ECCOLO;yes

```

A **ResetPlan** command run against the current production plan is stored in the plan audit log file as follows:

```

STAGEMAN|20080207|100758|20080207|110758|PLAN|DELETE|WK1|admin|
|/home/WK1/schedlog/M200803140127|
AWSBH025I The old Symphony file renamed /home/WK1/schedlog/M200803140127

```

Dynamic workload scheduling audit

Description

When you select the dynamic scheduling capability at installation time, the auditing feature is automatically installed. By default, the auditing feature is disabled.

Auditable events are as follows:

JobDefinitionAuditEvent

Maintains a track of operations performed on job definitions.

JobLogAuditEvent

Maintains a track of operations performed on job logs.

JobAuditEvent

Maintains a track of operations performed on jobs.

ResourceAuditEvent

Maintains a track of operations performed on resources.

RelationshipAuditEvent

Maintains a track of operations performed on relationships between resources.

RecoveryActionAuditEvent

Maintains a track of operations performed on recovery actions.

HistoryDataAuditEvent

Maintains a track of operations performed on historical data.

To configure the auditing of events, enable the auditing feature and optionally change the default values in the configuration file to define event types to be audited. The configuration file is located in the following path:

```
TWA_home\TDWB\config\audit.properties
```

Configuring the audit

Configure one or more of the properties in the `audit.properties` file to enable and configure auditing:

audit.enabled

Specifies whether the auditing feature is enabled or disabled. The default value is false. Supported values are as follows:

false

The auditing feature is not enabled.

true

The auditing feature is enabled.

onSecurityEnabled

The auditing feature is enabled if global security is enabled on WebSphere Application Server Liberty Base.

audit.consumer.file.auditFilePrefix

Specifies the file prefix for the auditing log file. The file name is defined using the file prefix plus the `_auditN.log` suffix, where `N` is a progressive number. If you want the date and time of the file creation specified in the file prefix, use the default format: `'tdwb_'yyyy-MM-dd`. For instance, using the default prefix `'tdwb_'yyyy-MM-dd` generates the `tdwb_2010-12-20_auditN.log` family of files. Note that the text between single quotation marks (') is not processed by the program and remains unchanged. This format creates a different file

for each day the auditing feature is enabled. Also, changing the prefix to 'tdwb_YYYY-MM' generates the `tdwb_2010-12_auditN.log` family of files. This format creates a different file for each month the auditing feature is enabled.

You can modify this format as required to create files on a weekly, monthly or yearly basis, depending on your auditing requirements. Depending on the date and time format you choose, the maximum size and number of log files vary. The maximum size and number of log files are defined using the `audit.consumer.file.maxFileSize` and `audit.consumer.file.maxAuditFiles` properties respectively. Use these three parameters to control the size of the audit logs stored. For example, using the default values for these parameters, then every day you will have a maximum of 10 MB x 100 files each day. Once the maximum is reached, the first file created is overwritten. If you want use less space to store audit logs, you can decided to change the maximum number of files or only have files on a monthly basis, by specifying the format for the `audit.consumer.file.auditFilePrefix` property as 'tdwb_YYYY-MM'.

audit.consumer.file.auditFileLocation

Specifies the path where the log files are created. The default path is `/audit`.

audit.consumer.file.maxFileSize

Specifies the maximum size in bytes of the log files. When a file reaches the maximum size, a new log file is created. The default value is 10000000 bytes (10 MB). This is also the highest supported value.

audit.consumer.file.maxAuditFiles

Specifies the maximum number of files with a specific prefix. When all files reach the maximum size and the maximum number of files is exceeded, the oldest file with a specific prefix is overwritten. The default value is 100 files. This is also the highest supported value.

Configuring dynamic audit events

The following table lists the supported actions and properties for each event with the related default values. You can configure these values in the `audit.properties` file.

Table 72. Auditable event properties

Event	Action	Property	Default value
JobDefinitionAuditEvent	create	<code>audit.tdwb.JobDefinitionAuditEvent.create.enabled</code>	true
	delete	<code>audit.tdwb.JobDefinitionAuditEvent.delete.enabled</code>	true
	get	<code>audit.tdwb.JobDefinitionAuditEvent.get.enabled</code>	true
	query	<code>audit.tdwb.JobDefinitionAuditEvent.query.enabled</code>	false
	update	<code>audit.tdwb.JobDefinitionAuditEvent.update.enabled</code>	true
JobLogAuditEvent	get	<code>audit.tdwb.JobLogAuditEvent.get.enabled</code>	true
JobAuditEvent	cancel	<code>audit.tdwb.JobAuditEvent.cancel.enabled</code>	true
	get	<code>audit.tdwb.JobAuditEvent.get.enabled</code>	true

Table 72. Auditable event properties (continued)

Event	Action	Property	Default value
	query	audit.tdwb.JobAuditEvent.query.enabled	false
	submit	audit.tdwb.JobAuditEvent.submit.enabled	true
ResourceAuditEvent	create	audit.tdwb.ResourceAuditEvent.create.enabled	true
	delete	audit.tdwb.ResourceAuditEvent.delete.enabled	true
	query	audit.tdwb.ResourceAuditEvent.query.enabled	false
	resume	audit.tdwb.ResourceAuditEvent.resume.enabled	true
	suspend	audit.tdwb.ResourceAuditEvent.suspend.enabled	true
	update	audit.tdwb.ResourceAuditEvent.update.enabled	true
	RelationshipAuditEvent	create	audit.tdwb.RelationshipAuditEvent.create.enabled
	delete	audit.tdwb.RelationshipAuditEvent.delete.enabled	true
	query	audit.tdwb.RelationshipAuditEvent.query.enabled	false
RecoveryActionAuditEvent	invoke	audit.tdwb.RecoveryActionAuditEvent.invoke.enabled	true
HistoryDataAuditEvent	move	audit.tdwb.HistoryDataAuditEvent.move.enabled	true

By default, auditing is disabled for query actions, while all the other actions are enabled. If the auditing feature is disabled, all properties are ignored.

Log file specifications

The elements used in the auditing log files are extensions to the Common Base Event (CBE) schema. The types and elements listed below are available in the auditing log files. Supported action types for each element are listed in [Table 72: Auditable event properties on page 375](#).

Action

Represents the action that is being taken. Each auditable event supports a different set of possible actions. See [Table 72: Auditable event properties on page 375](#). The Action type contains the following element:

Table 73. Elements in Action type

Element name	Element description	Always returned in the output
Action	The action type that is being taken on the dynamic workload broker object.	Yes

ObjectInfoList

Represents a list of dynamic workload broker objects. The `ObjectInfoList` type contains the following element:

Table 74. Elements in ObjectInfoList type

Element name	Element description	Always returned in the output
objectInfo	The class of the object being involved in the action	Yes

ObjectInfo

Represents information about a dynamic workload broker object in an `objectInfoList` type or in another `objectInfo` element. The `ObjectInfo` type contains the following elements:

Table 75. Elements in ObjectInfo type

Element name	Element description	Always returned in the output
objectClass	The class of the object being involved in the action.	Yes
objectName	The name of the dynamic workload broker object.	Only if available
objectNamespace	The namespace of the dynamic workload broker object.	Only if available
objectType	The type of the dynamic workload broker object.	Only if available
objectAlias	The alias of the dynamic workload broker object.	Only if available
objectIdentifier	The unique identifier of the dynamic workload broker object.	Only if available
objectRole	The role of the dynamic workload broker object, if any. For instance a Resource can have the source or destination role in a relationship	Only if available
objectSubmitterType	The type of the component which submitted the operation. The component is one of the following: <ul style="list-style-type: none"> • Dynamic Workload Broker Console • Command line • Dynamic workload broker workstation • Third party utility 	Only if available
objectInfo	A child <code>objectInfo</code> object. For instance, a relationship is always related to two resources.	Only if available

Outcome

Defines the outcome of a security event. The Outcome type contains the following elements:

Table 76. Elements in Outcome type

Element name	Element description	Always returned in the output
result	The status of the event. This information can be used when filtering the information in the log file.	Yes
failureReason	Additional information on the outcome of the operation.	Yes, if the operation was unsuccessful.

UserInfoList

Represents a list of `userInfo` elements, each representing the list of users in the delegation chain. The `UserInfoList` type contains the following element:

Table 77. Elements in UserInfoList type

Element name	Element description	Always returned in the output
objectInfo	An array of Information about each user in the delegation chain. The first <code>userInfo</code> element identifies the user which authenticated first. The last <code>userInfo</code> element identifies the user with whose credentials the action is being taken.	Yes

UserInfo

Represents information about a user. Elements of this type return information about the user involved in the operation being audited. The `UserInfo` type contains the following element:

Table 78. Elements in UserInfo type

Element name	Element description	Always returned in the output
UserInfo	The username provided to dynamic workload broker for authentication.	Yes

How to perform queries on log files

Log files can be very long and detailed. When you view your log files with the Log and Trace Analyzer, you can apply one or more queries to filter information in the file and make searches faster. You can use the following queries to filter only the relevant information or you can create your own queries depending on your requirements. The following queries are written in XPath query language.

- To filter all the events generated by a specific user:

```
/CommonBaseEvent [extendedDataElements/children[@name='userInfo' and values='username']]
```

- To filter all the events related to a specific object class:

```
/CommonBaseEvent [ extendedDataElements//children[@name='objectClass' and values='Resource']]
```

- To filter all the events related to a specific object:

```
//CommonBaseEvent [ extendedDataElements//children[@name='objectName' and values='myresource']/  
../children[@name='objectClass' and values='Resource']]
```

- To filter all the events related to a specific action:

```
/CommonBaseEvent [extendedDataElements[@name='action' and values='uninstall']]
```

- To filter all the events with SUCCESSFUL outcome:

```
/CommonBaseEvent [extendedDataElements/children[@name='result' and values='SUCCESSFUL']]
```

- The following query returns all create actions:

```
/CommonBaseEvent[ extendedDataElements[@name = 'action' and values = 'create']]
```

You can export this query into an XML file as follows:

```
<?xml version="1.0" encoding="UTF-8"?><cbeviewer_configuration>  
<logParserSets>  
  <logParserSet description="Parser for CBE log"  
    id="com.ibm.cbeviewer.parsers.cbeLogParserSet"  
    label="Common Base Event log"  
    parentId="com.ibm.cbeviewer.parsers.jdLogParserSet"/>  
  <logParserSet description="Parser for CEI Server"  
    id="com.ibm.cbeviewer.parsers.ceiLogParserSet"  
    label="Common Event Infrastructure server"  
    parentId="com.ibm.cbeviewer.parsers.jdLogParserSet"/>  
  <logParserSet description="Other parsers"  
    id="com.ibm.cbeviewer.parsers.otherParsersLogParserSet"  
    label="Other parsers"/>  
</logParserSets>  
<recent_expressions>  
  <xpath name="All Create Events">  
    /CommonBaseEvent[ extendedDataElements[@name = 'action' and values = 'create']]  
  </xpath>  
</recent_expressions></cbeviewer_configuration>
```

The following is a short example of a log file:

```
<CommonBaseEvent  
  creationTime="2007-06-06T14:26:23.311Z"  
  extensionName="TDWB_JOB_AUDIT_EVENT"  
  globalInstanceId="CEFC6DD156CA54D902A1DC1439E6EC4ED0"  
  sequenceNumber="1"  
  version="1.0.1">  
<extendedDataElements  
  name="userInfoList"  
  type="noValue">
```

```

    <children
      name="userInfo"
      type="string">
      <values>UNAUTHENTICATED</values>
    </children>
  </extendedDataElements>
  <extendedDataElements
    name="action"
    type="string">
    <values>submit</values>
  </extendedDataElements>
  <extendedDataElements
    name="outcome"
    type="noValue">
    <children
      name="result"
      type="string">
      <values>SUCCESSFUL</values>
    </children>
  </extendedDataElements>
</CommonBaseEvent>

```

Examples

The following examples describe a standard usage of the auditing feature.

In the following example, user `root` successfully retrieves the definition of a job named **MyTestJob** using the `jobstore` command.

```

<CommonBaseEvent
  creationTime="2007-06-21T16:05:19.455Z"
  extensionName="TDWB_JOB_AUDIT_EVENT"
  globalInstanceId="CE8F5E102AE3419AF7A1DC201135463A40"
  sequenceNumber="188"
  version="1.0.1">
  <extendedDataElements
    name="userInfoList"
    type="noValue">
    <children
      name="userInfo"
      type="string">
      <values>root</values>
    </children>
  </extendedDataElements>
  <extendedDataElements
    name="action"
    type="string">
    <values>get</values>
  </extendedDataElements>
  <extendedDataElements
    name="outcome"
    type="noValue">
    <children
      name="result"
      type="string">
      <values>SUCCESSFUL</values>
    </children>

```

```

</extendedDataElements>
<extendedDataElements
  name="objectInfoList"
  type="noValue">
  <children
    name="objectInfo"
    type="noValue">
    <children
      name="objectClass"
      type="string">
      <values>Job</values>
    </children>
    <children
      name="objectName"
      type="string">
      <values>MyTestJob</values>
    </children>
    <children
      name="objectIdentifier"
      type="string">
      <values>3ebf6d62-0b83-3270-9b83-83c393e9cbca</values>
    </children>
    <children
      name="objectSubmitterType"
      type="string">
      <values>TDWB CLI</values>
    </children>
  </children>
</extendedDataElements>
<extendedDataElements
  name="CommonBaseEventLogRecord:sequenceNumber"
  type="long">
  <values>80808</values>
</extendedDataElements>
<extendedDataElements
  name="CommonBaseEventLogRecord:threadID"
  type="int">
  <values>280</values>
</extendedDataElements>
<sourceComponentId
  application="JobManagement"
  component="None"
  componentIdType="Application"
  location="tdws08"
  locationType="Hostname"
  subComponent="None"
  threadId="Default : 84"
  componentType="http://www.ibm.com/namespace/autonomic/Tivoli_componentTypes"/>
<situation
  categoryName="ReportSituation">
  <situationType
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="ReportSituation"
    reasoningScope="INTERNAL"
    reportCategory="SECURITY"/>
  </situation>
</CommonBaseEvent>

```

In the following example, user `testuser` tries deleting a job instance named **MySecondJob** using the appropriate command line. The operation fails because the job was submitted by another user. Deleting jobs submitted by other users requires `Operator` or `Administrator` rights. For more information on access rights, see *IBM Workload Scheduler: Scheduling Workload Dynamically* or *IBM Workload Scheduler: Administration Guide*.

```
<CommonBaseEvent
  creationTime="2007-06-21T16:05:32.746Z"
  extensionName="TDWB_JOB_AUDIT_EVENT"
  globalInstanceId="CE8F5E102AE3419AF7A1DC20113D32BB20"
  sequenceNumber="189"
  version="1.0.1">
<extendedDataElements
  name="userInfoList"
  type="noValue">
  <children
    name="userInfo"
    type="string">
    <values>testuser</values>
  </children>
</extendedDataElements>
<extendedDataElements
  name="action"
  type="string">
  <values>cancel</values>
</extendedDataElements>
<extendedDataElements
  name="outcome"
  type="noValue">
  <children
    name="result"
    type="string">
    <values>UNSUCCESSFUL</values>
  </children>
  <children
    name="failureReason"
    type="string">
    <values>userNotAuthorized</values>
  </children>
</extendedDataElements>
<extendedDataElements
  name="objectInfoList"
  type="noValue">
  <children
    name="objectInfo"
    type="noValue">
    <children
      name="objectClass"
      type="string">
      <values>Job</values>
    </children>
    <children
      name="objectName"
      type="string">
      <values>MySecondJob</values>
    </children>
    <children
      name="objectIdentifier"
```

```

        type="string">
        <values>a05732c8-c008-3103-afd1-84b567d78de7</values>
    </children>
    <children
        name="objectSubmitterType"
        type="string">
        <values>TDWB CLI</values>
    </children>
</extendedDataElements>
<extendedDataElements
    name="CommonBaseEventLogRecord:sequenceNumber"
    type="long">
    <values>80964</values>
</extendedDataElements>
<extendedDataElements
    name="CommonBaseEventLogRecord:threadID"
    type="int">
    <values>292</values>
</extendedDataElements>
<sourceComponentId
    application="JobManagement"
    component="None"
    componentIdType="Application"
    location="tdws08"
    locationType="Hostname"
    subComponent="None"
    threadId="Default : 91"
    componentType="http://www.ibm.com/namespace/autonomic/Tivoli_componentTypes"/>
<situation
    categoryName="ReportSituation">
    <situationType
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="ReportSituation"
        reasoningScope="INTERNAL"
        reportCategory="SECURITY"/>
    </situation>
</CommonBaseEvent>

```

Keeping track of database changes using audit reports

To keep always track of the changes that impact objects stored in the database, you can use the following audit reports, which can be run in batch mode using the command line interface:

General audit report

The report provides information about objects that have been modified in the database. More specifically, it details who made the change, on which objects, and when.

Details report

The report provides further details about the changes implemented. It specifies who made the change, on which objects, when, and what has been changed. More specifically it shows the object definition before and after the change.

You can run these reports on DB2 and Oracle databases.

A sample business scenario

The administrator of an insurance company needs to keep track of all the changes impacting the insurance policies, conditions and terms of all the customers registered in the company database. To do it, the administrator periodically runs the audit general and details reports.

To satisfy this request, he creates an audit general report that provides details about which TWS objects have been modified in the database, who modified them and on which date. Then, to find out more details about the changes, he also creates an audit details report.

To accomplish his task, he runs the following steps:

1. He customizes the property files related to the audit reports, specifying the format and content of the report output.
2. He schedules jobs to obtain the reports:
 - a. The first job generates an audit to be saved locally.
 - b. The second job runs a detail report overnight to retrieve more details about the specific changes implemented. The report output is sent using an mail to the analyst. The information collected is used to keep all the insurance branch offices updated with any change and news.
3. The administrator adds the two jobs to a job stream scheduled to run weekly and generates the plan.

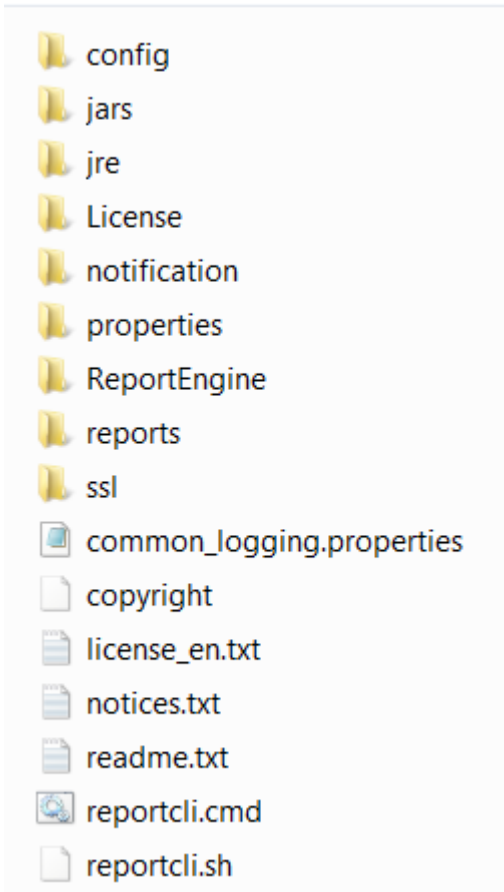
Setting up for command line audit reporting

About this task

Before running these reports you must perform a few setup steps:

1. The software needed to run these reports is contained in a package named `TWSBatchReportCli` included in the IBM Workload Scheduler installation image, in the `TWSBatchReportCli` directory. If you plan to run them from within a scheduled job, extract the package file on one of the operating systems listed at [Dynamic Workload Console Detailed System Requirements](#).

After extracting the package, you obtain the following file structure:



Because the native UNIX™ tar utility does not support long file names, if you are extracting the files on AIX®, Solaris, or HP-UX systems, ensure that the latest GNU version of tar (gtar) is installed to extract the files successfully.



Note:

- a. Make sure you run the following commands in the directory where you extracted the files:

On UNIX™

```
chmod -R +x *
chown -R username *
```

On Windows™

Ensure IBM Workload Scheduler is installed.

```
setown -u username *
```

Where *username* is the IBM Workload Scheduler user that will run the reports.

- b. If you plan to schedule jobs that run these reports, the system where you extract the package must be accessible as network file system from a fault-tolerant agent defined in the local scheduling environment.

2. If you use an Oracle database, download the JDBC drivers required by your Oracle server version.

3. Copy the JDBC drivers in the `report_cli_installation_dir\jars` directory and in `report_cli_installation_dir\ReportEngine\plugins` `\org.eclipse.birt.report.data.oda.jdbc_4.2.1.v20120820\drivers` directory. The report cli automatically discovers the two jar files.

4. Configure the template file `.\config\common.properties` by specifying the following information.

- a. If you use an Oracle database, connect to the database where the historical data are stored as follows:

- i. Retrieve the location of the Oracle JDBC drivers. This information is stored in the **com.ibm.tws.webui.oracleJdbcURL** property in the `TWSConfig.properties` file, located in

On Windows operating systems

`<TWA_home>\usr\servers\engineServer\resources\properties`

On UNIX operating systems

`<TWA_DATA_DIR>/usr/servers/engineServer/resources/properties`

For more information about this file, see [Configuring for an Oracle database on page 166](#).

- ii. Specify the location of the Oracle JDBC drivers in the **PARAM_DataSourceUrl** property in the `common.properties` file.

No customization is required if you use DB2.

- b. Set the date and time format, including the time zone. The file `.\config\timezone.txt` contains a list of time zones supported by IBM Workload Scheduler and the information on how to set them. The time zone names are case sensitive.

- c. Make the report output available on the URL specified in **ContextRootUrl** field. This is an example of the configuration settings:

```
#####
# HTTP Server information
#####

#Specify the context root where the report will be available
#To leverage this possibility it needs to specify in the report output dir
#the directory that is referred by your HTTP Server with this context root

ContextRootUrl=http://myserver/reportoutput
```

In this case, ensure that the `output_report_dir` specified when running the reports command points to the same directory specified in the **ContextRootUrl**.

- d. Send the report output using a mail. This is an example of the configuration settings:

```
#####
# Email Server configuration
#####

PARAM_SendReportByEmail=true

#SMTP server
mail.smtp.host=myhost.mydomain.com
#IMAP provider
mail.imap.socketFactory.fallback=false
mail.imap.port=993
mail.imap.socketFactory.port=993
#POP3 provider
```

```

mail.pop3.socketFactory.fallback=false
mail.pop3.port=995
mail.pop3.socketFactory.port=995

#####
# Email properties
#####
PARAM_EmailFrom=user1@your_company.com
PARAM_EmailTo=user2@your_company.com,user3@your_company.com
PARAM_EmailCC=user4@your_company.com
PARAM_EmailBCC=user5@your_company.com
PARAM_EmailSubject=Test send report by email
PARAM_EmailBody=This is the report attached

```

An explanation of all the customizable fields is contained in the template file.

Running audit reports from the command line

To run audit report on the database, you must first enable the audit feature and configure the audit options described in [Global options - detailed description on page 29](#).

The `\reports\templates` directory contains a sample template file for each type of report.

Before running any of these reports, ensure that you customize the corresponding template file, either `ad.properties` or `ag.properties`.

In that file, named `report_name.properties`, you can specify:

- The information to display in the report header.
- How to filter the information to display the expected result.
- The format and content of the report output.

For more information about the specific settings see the explanation provided in the template file beside each field.

After you set up the environment as it is described in [Setting up for command line audit reporting on page 384](#), and you configured report template file, use the following syntax to run the report:

reportcli -p *report_name.property*

[-o *output_report_dir*]

[-r *report_output_name*]

[-k *key=value*]

[-k *key=value*]

.....

where:

-p *report_name.property*

Specifies the path name to the report template file.

-o *output_report_dir*

Specifies the output directory for the report output.

-r report_output_name

Specifies the name of the report output.

-k key=value

Specifies the value of a settings. This value override the corresponding value, if defined, in the

`common.properties` file or in the `report_name.properties` file.

Examples

1. In this example the `reportcli.cmd` is run with the default parameter:

```
reportcli.cmd -p D:\ReportCLI\TWSReportCli\reports\templates\ag.properties
-r audit1
```

2. In this example the `reportcli.cmd` is run using the `-k` parameter to override the values set for **PARAM_DateFormat** in the `.\config\common.properties` file:

```
reportcli.cmd -p D:\ReportCLI\TWSReportCli\reports\templates\ag.properties
-r audit2 -k PARAM_DateFormat=short
```

3. In this example the `reportcli.cmd` is run using the `-k` parameter to override he format specified for the report output in the `.properties` file:

```
./reportcli.sh -p /TWSReportCli/REPCLI/reports/templates/ag.properties
-r audit3 -k REPORT_OUTPUT_FORMAT=html -k OutputView=charts
```



Note: If the report is run through a IBM Workload Scheduler job, the output of the command is displayed in the job output.

Collecting job metrics

You can run the following SQL queries on the Workload Scheduler data base to retrieve the number of jobs run by IBM Workload Scheduler over a period of time. One query determines the number of jobs run by specific workstations, while the other query determines the number of jobs run on the entire IBM Workload Scheduler domain. You can run the queries from the command line interface of your database or you can add them in the Dynamic Workload Console to create your custom SQL reports, as described in section creating a task to create custom SQL reports in *Dynamic Workload Console User's Guide*.

Job metrics queries for DB2

Use the following SQL query to find the number of jobs run on specific workstations:

```
SELECT year(job_run_date_time) AS Year, month(job_run_date_time) AS Month,
cast (count(job_run_date_time) AS INT) AS JobNbr FROM mdl.job_history_v
WHERE workstation_name IN ('WKS_1', 'WKS_2', 'WKS_N') or
(workstation_name = '-' and JOB_STREAM_WKS_NAME_IN_RUN in('WKS_1', 'WKS_2', 'WKS_N') )
GROUP BY year(job_run_date_time), month(job_run_date_time)
```

where 'WKS_1', 'WKS_2', 'WKS_N' are the names of the workstations that ran the jobs you want counted.

Use the following SQL query to find the number of jobs run on the entire IBM Workload Scheduler domain:

```
SELECT year(job_run_date_time) AS Year, month(job_run_date_time) AS Month,
cast (count(job_run_date_time) AS INT) AS JobNbr FROM mdl.job_history_v
GROUP BY year(job_run_date_time), month(job_run_date_time)
```

Job metrics queries for DB2 for zOS

Use the following SQL query to find the number of jobs run on specific workstations:

```
SELECT year(job_run_date_time) AS Year, month(job_run_date_time) AS Month,
cast (count(job_run_date_time) AS INT) AS JobNbr FROM mdl.job_history_v
WHERE workstation_name IN ('WKS_1', 'WKS_2', 'WKS_N')
GROUP BY year(job_run_date_time), month(job_run_date_time)
```

where 'WKS_1', 'WKS_2', 'WKS_N' are the names of the workstations that ran the jobs you want counted.

Use the following SQL query to find the number of jobs run on the entire IBM Workload Scheduler domain:

```
SELECT year(job_run_date_time) AS Year, month(job_run_date_time) AS Month,
cast (count(job_run_date_time) AS INT) AS JobNbr FROM mdl.job_history_v
GROUP BY year(job_run_date_time), month(job_run_date_time)
```

Job metrics queries for Oracle database

Use the following SQL query to find the number of jobs run on specific workstations:

```
SELECT EXTRACT(year FROM job_run_date_time) AS Year,
EXTRACT(month FROM job_run_date_time) AS Month,
cast (count(job_run_date_time) AS INT) AS JobNbr FROM job_history_v
WHERE workstation_name IN ('WKS_1', 'WKS_2', 'WKS_N')
or (workstation_name = '-' and JOB_STREAM_WKS_NAME_IN_RUN in('WKS_1', 'WKS_2', 'WKS_N'))
GROUP BY EXTRACT(year FROM job_run_date_time), EXTRACT(month FROM job_run_date_time);
```

where 'WKS_1', 'WKS_2', 'WKS_N' are the names of the workstations that ran the jobs you want counted.

Use the following SQL query to find the number of jobs run on the entire IBM Workload Scheduler domain:

```
SELECT EXTRACT(year FROM job_run_date_time) AS Year,
EXTRACT(month FROM job_run_date_time) AS Month,
cast (count(job_run_date_time) AS INT) AS JobNbr FROM job_history_v
GROUP BY EXTRACT(year FROM job_run_date_time), EXTRACT(month FROM job_run_date_time);
```

Chapter 8. Administrative tasks

This chapter describes how to perform some specific administrative tasks on IBM Workload Scheduler, as follows:

The tasks

[Switching a domain manager on page 391](#)

Change a domain manager or dynamic domain manager, either in the event of the failure of the computer where it is installed, or as part of a planned replacement activity.

[Switching the master to a backup on page 396](#)

Change a master domain manager, either in the event of the failure of the computer where it is installed, or as part of a planned replacement activity.

[Changing key IBM Workload Scheduler passwords on page 412](#)

Change the password of the <TWS_user>, or any other of the users that have an infrastructure role in IBM Workload Scheduler.

[Unlinking and stopping IBM Workload Scheduler on page 417](#)

The correct procedure to unlink the master domain manager from its agents and stop the master processing.

[Changing the properties for the database on page 418](#)

If you need to change the host, port or name of the database, effect the change in the application server, where the data source configuration is maintained.

[Changing the workstation host name or IP address on page 420](#)

Change the host name or IP address of a workstation.

[Changing the security settings on page 424](#)

If you need to update the properties that define your SSL connection or authentication mechanism, you need to make the changes in the WebSphere Application Server Liberty Base

[Managing the event processor on page 426](#)

If you are using event-driven workload automation, you will need to perform periodic maintenance on the event processor.

Application server tasks

The following tasks might need to be performed on the application server:

[Application server - starting and stopping on page 432](#)

How to stop and start the application server when you need to.

[Application server - automatic restart after failure on page 434](#)

The application server is managed by a utility that restarts it if it stops for any reason (subject to a configurable policy). This section describes how to modify the policy and deal with any situations that the policy cannot handle.

[Application server - encrypting the profile properties files on page 437](#)

Several of the application server configuration files contain passwords. To avoid that these remain in the files in plain text, run a utility to encrypt them.

[Application server - configuration files backup and restore on page 437](#)

The application server configuration manages the data source and security aspects of your IBM Workload Scheduler environment. The files should be regularly backed up and when necessary can be restored.

[Application server - changing the host name or TCP/IP ports on page 437](#)

If you need to change the host or ports used by the application server, follow the correct procedure.

[Application server - changing the trace properties on page 438](#)

The application server has a trace facility. This section describes how to increase the trace level to obtain more information for troubleshooting, and how to reduce the level to improve performance.

Changing the application server properties

Several of the above tasks require you to run a common procedure based on templates whereby you:

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.

This procedure is fully described in [Configuring IBM Workload Scheduler using templates on page 428](#).

Switching a domain manager

About this task

Being prepared for network problems makes recovery easier. Set up a backup domain manager for each domain manager in your network to more easily ensure that IBM Workload Scheduler peak job scheduling loads are met. Choose any fault-tolerant agent in the domain to be a backup domain manager.

A domain manager might need to be changed because you want it to run on a different workstation, or it might be forced on you as the result of network linking problems or the failure of the domain manager workstation itself. This section, and its subsections, describes how to prepare for and use a backup domain manager. However, if the domain manager to be

changed is a master domain manager or dynamic domain manager, there are some specific additional steps to perform; see [Switching the master to a backup on page 396](#).

Running without a domain manager has the following effects:

- Agents and subordinate domain managers cannot resolve inter-workstation dependencies, because activity records broadcast by the master domain manager are not being received.
- The upward flow of events is interrupted. This impacts events that report the status of jobs, job streams and dependencies defined on workstations in the IBM Workload Scheduler network hierarchy under the failed domain manager.
- Standard agents that are hosted by the failed domain manager cannot perform any processing, since they depend on the domain manager for all scheduling and job launching.

If the problem is expected to be of short duration, you can wait for the problem to be resolved and IBM Workload Scheduler will recover on its own, as described in the *Troubleshooting Guide* in the section about network linking problems. If you are uncertain about the duration, or if you want to restore normal agent operation, you must switch to a backup, as described in the following sections.

Ensure that the *FullStatus* mode is selected in the backup workstation definition. For more information about workstation properties, see the section about workstation definition in *User's Guide and Reference*.

Also ensure that the backup domain manager is synchronized with respect to time with the domain manager. The most secure way is to use a Network Time Protocol Server to control the time on both systems, with the same repeat interval.

Network security is enforced using IP address validation. As a consequence, workstation linking (autolink option or link command) might fail if an agent has an old *Symphony* file that does not contain the new domain manager. If a connection fails, remove the old *Symphony* file on the agent and retry the connection.

For more information about the autolink option, see the section about workstation definition in *User's Guide and Reference*.

For more information about the link command, see the section about the link command in *User's Guide and Reference*.

Simplified procedure for switching a domain manager

Use one of these procedures when you have a short-term loss of a domain manager.

Using the command line

See the procedure described under the `switchmgr` command in *User's Guide and Reference*.

Using the Dynamic Workload Console

1. In the navigation bar at the top, click **Monitoring and Reporting > Workload Monitoring > Monitor Workload**.
2. Select an engine.
3. In **Object Type**, select **Workstation**.
4. From the **Query** drop-down list, select a query to monitor workstations.

5. Click **Run** to run the monitoring task.
6. From the table containing the list of workstations, select a workstation and click **More Actions > Become Master Domain Manager**.

Domain managers remain switched until you perform another switch manager operation, or run JnextPlan. To return to the original domain manager without running JnextPlan, repeat this procedure.

Here is the procedure to follow every time you switch the master domain manager or dynamic domain manager if you run dynamic scheduling in your network:

1. Set the job fence to **go** priority level. For further details, see the fence command in *User's Guide and Reference*.
2. Switch the master domain manager or dynamic domain manager to a backup workstation. Use either the `conman switchmgr` command or the Dynamic Workload Console. For more information about both methods, see the procedure described under the `switchmgr` command in *User's Guide and Reference*.
3. Once the switch has been performed, restore the job fence to zero. For further details, see the fence command in *User's Guide and Reference*.

Complete procedure for switching a domain manager

This section summarizes the steps required to replace a running domain manager with its backup and to complete the procedure by restoring the original domain manager to its function. Follow these steps to make sure that no overlapping problems arise with obsolete versions of the Symphony file. You can also follow these steps to switch a master domain manager or a dynamic domain manager. The steps are documented for four scenarios:

Planned outage

The domain manager is replaced with its backup for planned maintenance work (for example, an upgrade of the operating system).

Unplanned outage

The domain manager is replaced with its backup because of an unexpected failure or malfunction.

Short-term

The domain manager is expected to return to service before the next new production period turnover (run of the JnextPlan job).

Long-term

The domain manager is not expected to return to service before the next new production period turnover (run of the JnextPlan job).

Table 79. Complete procedure for switching a domain manager in case of a planned outage.

Planned outage	
Short-term	Long-term
1. Switch the domain manager to a backup workstation. Use either the <code>conman switchmgr</code>	1 Switch the domain manager to a backup workstation. Use either the <code>conman switchmgr</code> command or the Dynamic Workload Console.

Table 79. Complete procedure for switching a domain manager in case of a planned outage. (continued)

Planned outage	
<p>command or the Dynamic Workload Console. For more information about both methods, see the <code>Switching a master domain manager or dynamic domain manager</code> chapter in the Administration Guide.</p> <p>Check that the message boxes for the domain manager undergoing maintenance are large enough not to fill up before it is restored. Increase their size if necessary.</p> <p>2. Shut down IBM Workload Scheduler processing on the domain manager undergoing maintenance.</p>	<p>For more information about both methods, see the <code>Switching a master domain manager or dynamic domain manager</code> chapter in the Administration Guide.</p> <p>Check that the message boxes for the domain manager undergoing maintenance are large enough not to fill up before it is restored. Increase their size if necessary.</p> <p>2. Shut down IBM Workload Scheduler processing on the original domain manager undergoing maintenance.</p> <p>3. In the IBM Workload Scheduler database assign the role of domain manager to the backup workstation. This is done by changing the workstation type in the database from MANAGER to FTA on the original domain manager and from FTA to MANAGER on the backup.</p> <p>4. Set the workstation running the original domain manager to <code>ignore</code>, using either the <code>composer mod cpu <workstation_name></code> command or the Dynamic Workload Console.</p> <p>5. Run JnextPlan to generate the new production plan so that the backup master domain manager is removed from the plan.</p> <p>When ready to restore the ownership of the domain to the original domain manager:</p> <p>6. Remove the <code>ignore</code> flag from the workstation running the original domain manager.</p> <p>7. Run JnextPlan to generate the new production plan so that the backup master domain manager is reinserted in the plan.</p> <p>8. Reassign ownership of the domain to the original domain manager in the IBM Workload Scheduler database. This is done by changing the workstation type in the database from MANAGER to FTA on the original backup and from FTA to MANAGER on the original domain manager</p> <p>Optionally, remove in the original domain manager the <code>conman start</code> command from the init procedure and delete any existing copies of the Symphony, Sinfonia, and message box files. This step is recommended to avoid that any outdated symphony present in the computer is</p>
<p>When ready to restore the ownership of the domain to the original domain manager:</p> <p>3. Switch from the backup workstation to the domain manager using one of the methods indicated in step 1.</p> <p>4. Link the domain manager from the master to download a fresh version of the Symphony file.</p>	

Table 79. Complete procedure for switching a domain manager in case of a planned outage. (continued)

Planned outage
<p>automatically triggered at the first startup. You can add <code>conman start</code> again later.</p> <p>9. Switch from the backup workstation to the domain manager using one of the methods indicated in step 1.</p> <p>10. Link the domain manager from the master to download a fresh version of the Symphony file.</p>

Table 80. Complete procedure for switching a domain manager after an unplanned outage.

Short-term	Long-term
<p>1. Switch the domain manager to a backup workstation. Use either the <code>conman switchmgr</code> command or the Dynamic Workload Console. For more information about both methods, see the <code>switchmgr</code> command in <i>User's Guide and Reference</i>.</p> <p>Check that the message boxes for the failing domain manager are large enough not to fill up before it is restored. Increase their size if necessary.</p> <p>When ready to restore the ownership of the domain to the original domain manager:</p> <p>Optionally, remove in the original domain manager the <code>conman start</code> command from the init procedure and delete any</p>	<p>1 Switch the domain manager to a backup workstation. Use either the <code>conman switchmgr</code> command or the Dynamic Workload Console. For more information about both methods, see the <code>switchmgr</code> command in <i>User's Guide and Reference</i>.</p> <p>Check that the message boxes for the failing domain manager are large enough not to fill up before it is restored. Increase their size if necessary.</p> <p>2. In the IBM Workload Scheduler database assign the role of domain manager to the backup workstation. This is done by changing the workstation type in the database from MANAGER to FTA on the original domain manager and from FTA to MANAGER on the backup.</p> <p>3. Set the workstation running the failing domain manager to <code>ignore</code>, using either the <code>composer mod cpu <workstation_name></code> command or the Dynamic Workload Console.</p> <p>5. Run JnextPlan to generate the new production plan so that the backup master domain manager is removed from the plan.</p> <p>When ready to restore the ownership of the domain to the original domain manager:</p> <p>4. Remove the <code>ignore</code> flag from the workstation running the original domain manager.</p>

Table 80. Complete procedure for switching a domain manager after an unplanned outage. (continued)**Unplanned outage**

existing copies of the Symphony, Sinfonia, and message box files. This step is recommended to avoid that any outdated symphony present in the computer is automatically triggered at the first startup. You can add `conman start` again later.

For an "unplanned outage", FTA needs a new Symphony file, on the current master domain manager (previous backup master domain manager) do the following:

1. Verify that it is linked to all agents except the old master domain manager
2. Shut down all IBM Workload Scheduler processes (unlink from all agents).
3. Rename Sinfonia as Sinfonia.orig
4. Copy Symphony to Sinfonia.orig

You now have identical Symphony and Sinfonia files.

2. Switch from the backup workstation to the domain manager using one of the methods indicated in step 1.
3. Link the domain manager from the master to download a fresh version of the Symphony file.

5. Reassign ownership of the domain to the original domain manager in the IBM Workload Scheduler database. This is done by changing the workstation type in the database from MANAGER to FTA on the original domain manager and from FTA to MANAGER on the backup.

Optionally, remove in the original domain manager the `conman start` command from the init procedure and delete any existing copies of the Symphony, Sinfonia, and message box files. This step is recommended to avoid that any outdated symphony present in the computer is automatically triggered at the first startup. You can add `conman start` again later.

7. Switch from the backup workstation to the domain manager using one of the methods indicated in step 1.

8. Link the domain manager from the master to download a fresh version of the Symphony file.

7. Run JnextPlan to generate the new production plan so that the backup master domain manager is reinserted in the plan.

Switching the master to a backup

About this task

A backup workstation for the master domain manager and another workstation for the dynamic domain manager are an essential asset to ensure business continuity and data integrity in your environment.

There are two ways in which the switchover to a backup master domain manager can occur:

A manual, planned switchover procedure

You can switch the master domain manager to a backup master domain manager at any time, either for a short term or for a long term (the original master is not expected to return to service before the next new production period turnover), using the `switchmgr` command.

An automatic failover process

Starting with version 9.5 Fix Pack 2, you can rely on the automatic failover feature, where, given a list of available backups, the workload is switched over to the backup. See [Automatic failover on page 399](#) for more information.

When selecting a workstation to be a backup, the same rules apply to both the automatic failover and the manual switching of the master. Backup workstations must have compatible operating systems with the master and the backup master domain manager must be installed on a system that is not currently defined in the workload scheduling network. For more information about these topics see [Selecting a workstation for the backup master domain manager on page 397](#) and [Changing an agent to become a backup master domain manager on page 398](#).

In the following topics you can find information about how to enable the automatic failover process, and how to manually switch a master domain manager, and a dynamic domain manager for a Z controller.

If you lose or want to plan to change a master domain manager or dynamic domain manager, the same comments in the section [Switching a domain manager on page 391](#) apply, but in addition, consider the sub-topics in this section.

Selecting a workstation for the backup master domain manager

It is the normal process to install a backup master domain manager when you set up your scheduling network. However, if you have not done so, and decide later that you need a backup master domain manager, you have two options:

- Install a backup master domain manager on a system that is not currently in the workload scheduling network. For the detailed procedure, see the section about installing the master domain manager and backup master domain manager *Planning and Installation Guide*.
- Promote an agent to backup master domain manager. This option is time-consuming and requires you to interrupt your workload scheduling activities, but if you want to do it, follow the procedure described in this section.

Regardless of the option you choose, the following are some prerequisites to consider for the backup workstation:

- Choose compatible operating systems. Since you must transfer files between the master domain manager and its backup, the workstations must have compatible operating systems. Do not combine UNIX™ with Windows™ workstations, and in UNIX™, do not combine big-endian workstations (HP-UX, Solaris, and AIX®) with little-endian workstations (most Intel™-based operating systems, including Windows™ and Linux™).

See the [IBM Workload Scheduler Detailed System Requirements](#) for details of the prerequisite requirements of a backup master domain manager.

- Ensure the master domain manager and the backup master domain manager have `FullStatus` turned on in the workstation definition. See [Setting up a backup master domain manager on page 398](#)
- Copy any necessary files, such as, the security file and localopts file, to the backup workstation. See [Copying files to use on the backup master domain manager on page 399](#).

Changing an agent to become a backup master domain manager

You *cannot* change an agent to become a backup master domain manager, using a command or procedure that allows continuity of workload scheduling activities.

Instead, if you need to change an agent workstation to become the backup master domain manager, you must interrupt the workload scheduling activities. The procedure is as follows:

1. Check that the workstation satisfies the prerequisites for a backup master domain manager. See [IBM Workload Scheduler Detailed System Requirements](#) for more information.
2. If it does, stop and disable all workload scheduling operations on the workstation
3. Uninstall the agent, following the instructions in the section about uninstalling agents in *Planning and Installation Guide*.
4. Install the backup master domain manager on the system where the agent was installed, following the instructions in the section about installing the master domain manager and backup master domain manager in *Planning and Installation Guide*.
5. Ensure that the database entry for the workstation is correct for a backup master domain manager. See the section about workstation definition in *User's Guide and Reference* for information about the workstation definition
6. Define and start any workload scheduling operations you require on the workstation in its new role.

Setting up a backup master domain manager

Ensure that the master domain manager and the backup master domain manager have `FullStatus` turned on in the workstation definition. This is important if you need to resort to long-term recovery, where the backup master domain manager generates a `Symphony` file (runs `JnextPlan`). If `FullStatus` is not turned on, the former master domain manager shows up as a regular fault-tolerant agent after the first occurrence of `JnextPlan`. During normal operations, the `JnextPlan` job automatically turns on the `FullStatus` flag for the master domain manager, if it is not already turned on. When the new master domain manager runs `JnextPlan`, it does not recognize the former master domain manager as a backup master domain manager unless the flag is enabled. The former master domain manager does not have an accurate `Symphony` file when the time comes to switch back. For more information about workstation properties, see the section about workstation definition in *User's Guide and Reference*.

Also ensure that the backup master domain manager is synchronized with respect to time with the master domain manager. The securest way is to use a Network Time Protocol Server to control the time on both systems, with the same repeat interval.

Copying files to use on the backup master domain manager

To back up the important master domain manager files to the backup master domain manager, use the following procedure:

1. Copy the `Security` file from the master domain manager to backup domain manager in the following path:

On Windows operating systems

`<TWA_home>\TWS`

On UNIX operating systems

`TWA_DATA_DIR`

Add a suffix to the file so that it does not overwrite the `Security` file on the backup domain manager, for example, `Security_from_MDM`.

2. Copy all files in the following path:

On Windows operating systems

`<TWA_home>/TWS/mozart`

On UNIX operating systems

`TWA_DATA_DIR/mozart`

3. Copy the `localopts` file (see [Setting local options on page 51](#) for the location). Add a suffix to the file so that it does not overwrite the `localopts` file on the backup master domain manager; for example, `localopts_from_MDM`.

This procedure must be performed each production period, or whenever there are significant changes to any objects. It can be incorporated into a script.

In addition to these required files, you might also want to copy the following:

- Any scripts you might have written.
- Archived Symphony files, for reference.
- Log files, for reference.



Note: Another approach could be to place all of the above files on a separately mountable file system, that could easily be unmounted from the master domain manager and mounted on the backup master domain manager in the event of need. You would almost certainly want to backup these files in addition, to protect against loss of the separately mountable file system.

To prevent the loss of messages caused by a master domain manager error, you can use the fault-tolerant switch-manager facility.

Automatic failover

Switching a master domain manager to a backup master domain manager.

Recovery is easy when you are prepared for potential problems. If the master domain manager becomes unavailable, to ensure continuous operations, a long-term switchmgr operation is triggered and the workload is automatically switched to an eligible backup master domain manager. Similarly, the backup event processors automatically detect if the event processor is unavailable, and a long-term switcheventprocessor command triggered. This is the default behavior for a complete fresh installation of V9.5 Fix Pack 2 or later, but it can be enabled for back-level environments that are upgraded to V9.5 Fix Pack 2 or later.



Note: If you perform a fresh installation of a backup master domain manager at the V9.5 Fix Pack level in an existing back-level environment, the automatic failover feature is disabled. To enable it, follow this procedure. The feature is enabled by default for only a complete fresh installation.

You can optionally define potential backups for both the master domain manager and the event processor in two separate lists, adding preferential backups at the top of the lists. The backup engines monitor the behavior of the master domain manager and event processor to detect anomalous behavior and then attempt to recover. Each component plays a role in either detecting a failure or recovering from it:

- Each backup master domain manager monitors the status of the active master domain manager.
- The master domain manager (active or backup) is made to be self-aware. It monitors the status of its fault-tolerant agent to check on the status of processes such as, Batchman, Mailman and Jobman. If at least one of these processes are down, the master domain manager makes 3 attempts to restart them.
- If the WebSphere Application Server Liberty Base goes down, the watchdog process attempts to restart it.
- If the active master domain manager cannot be automatically restored within 5 minutes (the threshold after which the master is declared unavailable), then a permanent switch to a backup is automatically triggered by any of the backup candidates when one or more of the following conditions persist:
 - The fault-tolerant agent, WebSphere Application Server Liberty Base, or both are still down.
 - The engine is unable to communicate with the database, for example, due to a network outage.

If you have defined potential backups in a list, and a switch after 5 minutes is not possible with the first backup in the list because it is unavailable, then an attempt is made to contact the remaining backups in the list, following the order specified in the list, until an available backup is found to perform the switch. In this case, 5 minutes pass between each attempt.

The list for potential event processor backups is a separate list from the potential master domain manager backups, because you might have a workstation that can serve as the event manager backup, but you do not want it to act as a potential master domain manager backup. If the event manager fails, but the master domain manager is running fine, then only the event manager switches to a backup manager defined in the list of potential backups.

You can track detected failures and the actions taken by checking the `messages.log` file located in the path:

- `<TWA_DATA_DIR>/stdlist/appserver/engineServer/logs/messages.log`
- `<TWA_home>\TWS\stdlist\appserver\engineServer\logs\messages.log`



Note: On Linux® and UNIX®, for a fresh installation, an extended agent is installed with the master domain manager which is used to communicate where to run the FINAL job stream, along with its jobs. With an extended agent, \$MASTER can be used to indicate that the agent's host workstation is the master domain manager. If the role of the master is switched to a backup, then the new master is represented by \$MASTER. This supports both a short-term and long-term switch for the automatic failover feature. If you are upgrading from a version earlier than 9.5 Fix Pack 2, then you must define the extended agent manually.

On Windows™ workstations, the FINAL job stream is not defined on the extended agent, but remains on the master domain manager. The FINAL and FINALPOSTREPORTS job streams and jobs need to be moved from the master to the extended agent workstation. For this reason, only a short-term switch can be performed automatically and the long-term switch must be performed manually as documented in [Extended loss or permanent change of master domain manager on page 406](#) and in [Complete procedure for switching a domain manager on page 393](#). See also the switchmgr command in the *User's Guide and Reference* that contains both the command-line syntax, as well as the procedural steps to perform the switch from the Dynamic Workload Console.

Enabling automatic failover

To enable automatic failover, configure one or more backup engines, and set the related global options using the optman command, so that when the active master becomes unavailable, a long-term switchmgr operation is triggered.

Before you begin

Ensure that the master domain manager and the back master domain managers were installed using the same user (UID) and group (GID).

About this task

If you performed an upgrade from Version 9.5 or 9.5 Fix Pack 1, the automatic failover feature is disabled, but it can be enabled following a few simple steps outlined in this task. Automatic failover is, instead, enabled by default for a fresh installation of Version 9.5 Fix Pack 2 and later, and any backup master domain manager installed and configured with Fix Pack 2 is an eligible backup. If you subsequently disabled this feature, you can use the following procedure to re-enable it. You can also use this procedure to define a list of preferred eligible backups, excluding any backups you do not want to consider as an eligible backup. You can also configure a separate list of potential backups for the event processor.

1. Ensure the local option, mm resolve master, in the localopts file, is set to no on both the master domain manager and on all eligible backup master domain managers.
2. Optional. Define a list of potential backups for the master domain manager and the event manager.
 - a. Update the global option, workstationMasterListInAutomaticFailover, on the master domain manager to specify a list of workstations to be considered as eligible backups for the master domain manager. Edit the value of this option by adding a list of workstations, separated by commas, starting with your preferred choices at the top of the list. The list includes the current master domain manager. If no workstations are specified in this list, then the first backup master domain manager to detect that the master is down, performs the switch.

- b. Specify potential backups for the event processor by editing the value for the workstationEventMgrListInAutomaticFailover global option. Add a list of workstations, separated by commas, starting with your preferred choices at the top of the list. The list includes the current event manager workstation.
3. Set the following global options to "yes": enAutomaticFailover | af and enAutomaticFailoverActions | aa using the optman chg command. For example:

```
optman chg af=yes
optman chg aa=yes
```

4. Restart WebSphere Application Server Liberty Base



Important: Complete the remaining steps only if they are not already present in your environment.

5. If not already present on the master domain manager, create a new workstation with the following specifications:
- **Type:** Extended Agent
 - **Access method:** unixlocl
 - **Host:** \$MASTER

For example, if you create a workstation named, MDM_XA, with these specifications, the following is the workstation definition:

```
CPUNAME MDM_XA
DESCRIPTION "Workload Scheduler Virtual Master"
OS OTHER
NODE mdm_xa TCPADDR 31111
FOR MAESTRO HOST $MASTER ACCESS "unixlocl"
TYPE X-AGENT
AUTOLINK OFF
BEHINDFIREWALL OFF
FULLSTATUS OFF
END
```

6. Set the FINAL and FINALPOSTREPORTS job streams on the master domain manager to "draft". Draft job streams are not added to the preproduction plan.

```
composer mod js=FINAL
composer mod js=FINALPOSTREPORTS
```

For example, the following is an extract from the definition for the FINAL job stream:

```
SCHEDULE MDM#FINAL
DESCRIPTION "Added by composer."
DRAFT
ON RUNCYCLE RC1 "FREQ=DAILY;"
AT 2359
CARRYFORWARD
FOLLOWS MDM#FINAL.SWITCHPLAN PREVIOUS
:
```

The following example is an extract from the definition for the FINALPOSTREPORTS job stream:

```
SCHEDULE MDM#FINALPOSTREPORTS
DESCRIPTION "Added by composer."
DRAFT
ON RUNCYCLE RC1 "FREQ=DAILY;"
```

```
SCHEDTIME 2359
CARRYFORWARD
FOLLOWS MDM_XA#FINAL.SWITCHPLAN PREVIOUS
:
```

7. If not already present, make the following changes to the `Sfinal` file:

a. Create a backup of the `Sfinal` file. For example:

```
cp /<TWA_home>/TWS/Sfinal /<TWA_home>/TWS/Sfinal.orig
```

b. Add the new extended agent workstation to the FINAL and FINALPOSTREPORTS job stream definitions.

c. Substitute the SCRIPTNAME keyword with DOCOMMAND in all of the jobs defined in the FINAL and FINALPOSTREPORTS job streams.

d. Ensure the path to the scripts launched by the jobs in the FINAL and FINALPOSTREPORTS job streams use the variable, `UNISONHOME`.

e. Submit the `composer add Sfinal` command to generate the FINAL and FINALPOSTREPORTS job streams on the new extended agent workstation if they do not already exist.

f. Verify that the new extended agent workstation has been added to the `Sfinal` file. The following example is an extract of the modified `Sfinal` file containing the addition of the MDM_XA extended agent workstation, the substitution of the SCRIPTNAME keyword with DOCOMMAND in all jobs defined in FINAL and FINALPOSTREPORTS job stream definitions, and the use of the `UNISONHOME` variable in place of the path to the scripts:

FINAL:

```
SCHEDULE MDM_XA#FINAL ON EVERYDAY
        AT 2359
        CARRYFORWARD
FOLLOWS MDM_XA#FINAL.SWITCHPLAN PREVIOUS
...
...
...
        STARTAPPSERVER DOCOMMAND
"#{UNISONHOME}/../appservertools/startAppServer.sh"
        STREAMLOGON wa95ids
        RECOVERY CONTINUE
        MAKEPLAN DOCOMMAND "#{UNISONHOME}/MakePlan"
        STREAMLOGON wa95ids
        RCCONDSUCC "(RC=0) OR (RC=4)"
        FOLLOWS STARTAPPSERVER
        SWITCHPLAN DOCOMMAND "#{UNISONHOME}/SwitchPlan"
        STREAMLOGON wa95ids
        FOLLOWS MAKEPLAN
...
...
...
END
```

FINALPOSTREPORTS:

```

SCHEDULE  MDM_XA#FINALPOSTREPORTS ON EVERYDAY
          SCHEDTIME 2359
          CARRYFORWARD
FOLLOWS  MDM_XA#FINAL.SWITCHPLAN  PREVIOUS
...
...
...
          CHECKSYNC  DOCOMMAND "${UNISONHOME}/bin/planman checksync"
          STREAMLOGON wa95ids
          RECOVERY CONTINUE
          CREATEPOSTREPORTS  DOCOMMAND "${UNISONHOME}/CreatePostReports"
          STREAMLOGON wa95ids
          RECOVERY CONTINUE
          UPDATESTATS  DOCOMMAND "${UNISONHOME}/UpdateStats"
          STREAMLOGON wa95ids
          RECOVERY CONTINUE
          FOLLOWS CHECKSYNC
...
...
...
END

```

8. Submit the `composer add Sfinal` command and then verify that the FINAL and FINALPOSTREPORTS job streams and the related jobs, are correctly defined on the extended agent workstation. The following is an example of the correct output:

```

...
...
...
/
-add Sfinal
AWSJCL003I The command "add" completed successfully on object "jd=MDM_XA#STARTAPPSERVER".
AWSJCL003I The command "add" completed successfully on object "jd=MDM_XA#MAKEPLAN".
AWSJCL003I The command "add" completed successfully on object "jd=MDM_XA#SWITCHPLAN".
AWSJCL003I The command "add" completed successfully on object "js=MDM_XA#FINAL".
AWSJCL003I The command "add" completed successfully on object "jd=MDM_XA#CHECKSYNC".
AWSJCL003I The command "add" completed successfully on object "jd=MDM_XA#CREATEPOSTREPORTS".
AWSJCL003I The command "add" completed successfully on object "jd=MDM_XA#UPDATESTATS".
AWSJCL003I The command "add" completed successfully on object "js=MDM_XA#FINALPOSTREPORTS".
AWSBIA090I For file "Sfinal": errors 0, warnings 0.
AWSBIA288I Total objects updated: 8

```

9. Compare the two copies of the FINAL and FINALPOSTREPORTS job streams and make any necessary changes to those on the extended agent workstation, for example, the job stream submit time, run cycles, or any other custom changes to personalize the schedule.
10. Submit JnextPlan with the `-noremove` options to update the plan with the new extended agent workstation:

```
JnextPlan -for 0000 -noremove
```

11. If JnextPlan runs correctly, proceed to delete the FINAL and FINALPOSTREPORTS job streams previously set to "draft" on the master domain manager.

```
composer del FINALPOSTREPORTS
composer del FINAL
```

12. Delete the FINAL and FINALPOSTREPORTS job streams from the plan as follows.

```
conman "canc FINALPOSTREPORTS"
conman "canc FINAL"
```

13. Modify the new job stream definitions for the FINAL and FINALPOSTREPORTS job streams, setting the limit to "0":

```
SCHEDULE MDM_XA#FINAL
DESCRIPTION "Added by composer."
ON RUNCYCLE RC1 "FREQ=DAILY;"
AT 2359
CARRYFORWARD
FOLLOWS MDM_XA#FINAL.SWITCHPLAN PREVIOUS
LIMIT 0
:
MDM_XA#STARTAPPSERVER
```

14. Submit first the FINAL, and then the FINALPOSTREPORTS job streams into the current plan.

```
conman sbs MDM_XA#FINAL
conman sbs MDM_XA#FINALPOSTREPORTS
```

15. Verify that the start time and date for the FINAL and FINALPOSTREPORTS job streams are correct by submitting the `conman showschedules` command.
16. Reset the value of the limit job stream keyword for the FINAL and FINALPOSTREPORTS job streams, both in the database and in the plan.

```
conman "limit MDM_XA#FINAL ;10"
conman "limit MDM_XA#FINALPOSTREPORTS ;10"
```

Both job streams should be in WAITING (HOLD internal status), awaiting execution time.

17. Archived plans, forecast and trial plans are stored on the master domain manager where the plans run. To make these plans available on the backup master domain manager, either store the plan in a single shared folder, or create a job that synchronizes the plans between the master domain manager and the backup master domain manager.

What to do next

To enable the new master to access the plans that ran on the original master (the current plan is visible because it is synchronized with the backup), configure a job that copies the plans from the original master to the new master.

After an automatic failover, if you would like to subsequently return service to the original master, you must perform a manual switch. See [Manually switching the master on page 405](#).

Manually switching the master

A manual switchover from the primary master domain manager to a backup master domain manager is invoked through the `switchmgr` command. The backup master domain manager becomes the current, active master connected to the IBM® Workload Scheduler database.

There are four main use case scenarios that can prompt the need for switch of the master to a backup:

Planned outage

The domain manager is replaced with its backup for planned maintenance work (for example, an upgrade of the operating system).

Unplanned outage

The domain manager is replaced with its backup because of an unexpected failure or malfunction.

Short-term

The domain manager is expected to return to service before the next new production period turnover (run of the JnextPlan job).

Long-term

The domain manager is not expected to return to service before the next new production period turnover (run of the JnextPlan job).

Short-term switch of a master domain manager

Use the procedure described in [Simplified procedure for switching a domain manager on page 392](#) when you have a short-term loss of a master domain manager.

Master domain managers remain switched until you perform another switch manager operation. To return to the original master domain manager, repeat this procedure before the next production period turnover, unless you do not expect the master domain manager to be available for the next production period turnover (final Job Scheduler and JnextPlan job). In this case, use the procedure in the following section.

Extended loss or permanent change of master domain manager

Use the following procedure to switch to the backup if the original master domain manager is not expected to return to service before the next new production period turnover (final Job Scheduler and JnextPlan job).

On UNIX™ operating systems, use forward slashes in path names.

1. Use the conman **stop** function to stop IBM Workload Scheduler on the master domain manager and its backup. for more information about the command, see the section about the stop command in *User's Guide and Reference*.
2. If you copied the `Security` file from the master domain manager to the backup master domain manager with a `suffix`, now delete the `Security` file on the backup master domain manager and rename the `Security` file with the `suffix` as just `Security`.
3. If you copied the `localopts` file from the master domain manager to the backup master domain manager with a `suffix`, now merge the `localopts` file on the backup master domain manager with the `localopts` file from the master domain manager. Look at each property in turn and determine which version you want to keep on what is going to be your new master domain manager. For example, the property `thiscpu` needs to be the one from the backup master domain manager, but the options for controlling how the processes run can be taken from the master domain manager.
4. On the backup master domain manager cancel the `final` Job Scheduler in the Symphony file (it refers to the next production period's JnextPlan on the old master domain manager).
5. On the backup master domain manager, use `composer` to modify any important job streams that run on the master domain manager, in particular the `final` Job Scheduler. For each of these, change the workstation name to the name of the backup.

6. Change the workstation definition of the master domain manager from `manager` to `fault-tolerant agent`.
7. Change the workstation definition of the backup master domain manager from `fault-tolerant agent` to `manager`.



Note: These two steps must be done in the order given, as the system will not allow you to have two managers at the same time.

8. On the backup master domain manager, edit the `TWA_home/TWS/mozart/globalopts` file and change the `master` option to the name of the backup master domain manager workstation (this is used mainly for reports production)
9. Use the conman **switchmgr** function to switch to the backup master domain manager. See [Simplified procedure for switching a domain manager on page 392](#).
10. Submit a new *final* Job Scheduler to the new master domain manager (old backup master domain manager).
11. Run `JnextPlan -for 0000` on the new master domain manager to generate the new `Symphony` file.
12. Remember to log on to the backup master domain manager when opening the Dynamic Workload Console, first defining a new engine to access it.
13. If the old master domain manager has failed or is being replaced, you can now delete its workstation definition and remove it from the network.

Switching a master domain manager or dynamic domain manager

Switching a master domain manager or dynamic domain manager affects the running dynamic workload broker server.

The installation of a master domain manager or dynamic domain manager and of its backup workstations includes also the installation of a dynamic workload broker server.

You might have to switch the master domain manager or dynamic domain manager because, for example, the system running the current workstation is down. When a conman `switchmgr` command is submitted, an automatic process is triggered by which the old server stops the dynamic scheduling services and the new server starts a new instance of the dynamic workload broker server when the older server has completed the switch. This process ensures that there is only one active dynamic workload broker server running at a time.

You can configure this automatic switch broker process on instances at version 9.5 or later, by modifying the following properties contained in the `SwitchBroker.properties` file located in `TWA_DATA_DIR/broker/config/SwitchBroker.properties`:

Table 81. Configurable properties for automatic switch broker process

Property	Description
Master.Switch.HostName	The master domain manager name used to identify the active master.
Master.Switch.ExpiringTime	The number of seconds the new master waits before becoming the active master. The default value is 300 seconds.

Table 81. Configurable properties for automatic switch broker process (continued)

Property	Description
Master.Switch.PollingTime	The time interval, in seconds, between the database checks made by the new master on the status updates of the old master. The default is 5 seconds.

When the `switchmgr` command is submitted, the new master begins monitoring the database (with the frequency specified by **Master.Switch.PollingTime**), to verify when the state changes for the old master. If there is no response from the old master (because of a crash or because the old master is at a product level version earlier than V9.5) then the new master waits for a maximum of two intervals specified by **Master.Switch.PollingTime** (10 seconds), and then automatically promotes itself as the new active master. If, instead, a status update is detected in the database while polling, the new master waits the amount of time specified by **Master.Switch.ExpiringTime** before declaring itself the new active master. As soon as the old master completes the switch procedure, then the new master declares itself as the active master without waiting for the expiry of the **Master.Switch.ExpiringTime**.

The properties must be modified on both the master domain manager or dynamic domain manager and their backups at version 9.5 or later.

Here is the procedure to follow every time you switch the master domain manager or dynamic domain manager if you run dynamic scheduling in your network:

1. Set the job fence to **go** priority level. For further details, see the section about the fence command in *User's Guide and Reference*.
2. Switch the master domain manager or dynamic domain manager to a backup workstation. Use either the `conman switchmgr` command or the Dynamic Workload Console. For more information about both methods, see the procedure described under the `switchmgr` command in *User's Guide and Reference*.
3. Once the switch has been performed, restore the job fence to zero. For further details, see the procedure described under the `switchmgr` command in *User's Guide and Reference*.

Switching a dynamic domain manager for a Z controller

As a normal behavior, the dynamic domain manager for a Z controller works by having both the server and processes up and running, while the backup dynamic domain manager for a Z controller works with the processes that are running and the system that is down. You might have to switch the dynamic domain manager for a Z controller to a backup workstation because, for example, the system running the current workstation goes down.

To switch the dynamic domain manager for a Z controller to the backup workstation, perform the following procedure:

1. Stop the server on the dynamic domain manager for a Z controller by issuing the following command:

On Windows

```
stopAppServer.bat
```


On UNIX

```
stopAppServer.sh
```

2. On the backup dynamic domain manager for a Z controller, run the following command:

On Windows

```
startZosDDM.bat -dbUsr db_user -dbPwd db_user_pwd
```

On UNIX

```
startZosDDM.sh -dbUsr db_user -dbPwd db_user_pwd
```

where:

dbUsr db_user

The user that has been granted access to the IBM® Z Workload Scheduler tables on the database server.

dbPwd db_user_pwd

The password for the user that has been granted access to the IBM® Z Workload Scheduler tables on the database server.

3. On the backup dynamic domain manager for a Z controller, start the server by running the following command:

On Windows

```
startAppServer.bat
```

On UNIX

```
startAppServer.sh
```

Cloning scheduling definitions from one environment to another

About this task

This section applies to IBM Workload Scheduler master domain managers and its backup. It documents how to clone IBM Workload Scheduler data from one environment to another.



Note: This cloning procedure does not clone the following information from the source environment:

- The preproduction plan
- The history of job runs and job statistics
- The audit records



- The state of running event rule instances. This means that any complex event rules, where part of the rule has been satisfied prior to cloning of the environment, are generated as new rules after the cloning procedure. Even if the subsequent conditions of the event rule are satisfied, the record that the first part of the rule was satisfied is no longer available, so the rule will never be completely satisfied.

With the following steps all scheduling object definitions and global options can be migrated from a source environment named "ENV_1" to a target environment named "ENV_2".

1. In ENV_2, install a fresh instance of an IBM Workload Scheduler version 9.5 master domain manager and it point to its database by defining `MDM_ENV2` as the master domain manager workstation name. The installation process automatically defines the following workstations in the `ENV_2` database:
 - `MDM_ENV2` is the master domain manager workstation name.
 - `MDM_ENV2_DWB` is the broker workstation name.
 - `MDM_ENV2_1` is the agent workstation name.
 - `MASTERAGENTS` is the dynamic pool which includes, by default, `MDM_ENV2_1` dynamic agent workstation.
2. On the ENV_1 master domain manager, run the `dataexport` command or script to export all scheduling object definitions and global options from `ENV_1`. You can find this file in the `bin` subdirectory of the `TWA_home` directory.

Run `dataexport` from a Windows™ or UNIX® command prompt as follows:

```
dataexport source_dir export_dir
```

where:

source_dir

The is the `TWS_HOME` directory of the `ENV_1` instance of IBM Workload Scheduler version 9.5.

export_dir

The directory where the export files are created. Ensure the `twuser` user has write rights on this directory.

For example:

```
dataexport.cmd F:\IWS95\twsDB2user F:\IWS95\export
```

The object definitions and the global options are retrieved from the `ENV_1` database and placed in the `F:\IWS95\export` directory.

3. Verify that the following files were created in `export_dir`:
 - `acls.def`
 - `calendars.def`
 - `erules.def`
 - `folders.def`
 - `globalOpts.def`
 - `jobs.def`



Note: The record length supported by DB2® is 4095 bytes, but it decreases to 4000 bytes with Oracle. When you migrate your job definitions to Oracle, any job with task string (scripts or commands) exceeding 4000 bytes in length are not migrated. In this case, the data import utility replaces the job definition with a dummy job definition and sets the job priority to 0, guaranteeing that successors are not run.

- `parms.def`
- `prompts.def`
- `rcgroups.def`
- `resources.def`
- `scheds.def`
- `sdoms.def`
- `srols.def`
- `topology.def`
- `users.def` (includes encrypted user passwords)
- `variables.def`

4. Open the `export_dir\topology.def` file and remove the `MASTERAGENTS` definition to avoid replacing the same workstation definition that was created when you installed a fresh instance of the IBM Workload Scheduler version 9.5 master domain manager in `ENV_2`.

If you plan to dismiss `ENV_1` and you want to move the other workstations from `ENV_1` to `ENV_2`, then you do not have to perform any additional steps.

If you plan to install new agents in `ENV_2`, then it is recommended to install them using the same **displayname** used by the agent present in `ENV_1` so that you do not need to modify the `erules.def`, `jobs.def`, `resources.def`, `scheds.def`, and `users.def` files exported in the previous step. If you do not install them using the same **displayname**, then you must edit these files so that they match the agent **displayname** present in `ENV_2`.

5. Edit the `export_dir\users.def` file to specify the current valid password for the Windows™ users.
6. To import the object definitions and the global options retrieved from the `ENV_1` into the `ENV_2` database, copy the files that were created when you ran the `dataexport` utility to a directory on the `ENV_2` master domain manager and run the `dataimport` command or script to import all scheduling object definitions and global options to the `ENV_2` database. You can find this file in the `bin` subdirectory of the `TWA_home` directory.

Run `dataimport` from a Windows™ or UNIX® command prompt as follows:

```
dataimport source_dir export_dir
```

where:

source_dir

The `TWS_HOME` directory of the `ENV_2` instance of IBM Workload Scheduler version 9.5.

export_dir

The directory where you copied the object definitions and the global options retrieved from `ENV_1`.

For example:

```
dataimport.cmd F:\IWS95\twsDB2user F:\IWS95\export
```

7. If you want to dismiss the instance of IBM Workload Scheduler installed in `ENV_1` and reuse the same agents in `ENV_2`, stop the IBM Workload Scheduler processes running on the master domain manager in `ENV_1` to avoid conflicts.

Results

You have now completed cloning scheduling definitions from one environment to another.

Changing key IBM Workload Scheduler passwords

About this task

When you change passwords for key users in your IBM Workload Scheduler environment, there are various operations to perform, depending on which user's password is being changed, the type of operating system on which it is deployed, and the type of IBM Workload Scheduler node where the password is being changed.

If you decide to proceed manually, the following pages describe what you have to do if the passwords of any of the following users change:

IBM Workload Scheduler instance owner

The `<TWS_user>` (the instance owner) of a IBM Workload Scheduler component .

WebSphere Application Server Liberty Base user

The WebSphere Application Server Liberty Base user which authenticates the `<TWS_user>` being used by IBM Workload Scheduler components. For more information, see the WebSphere Application Server Liberty Base documentation, for example [securityUtility command](#).

This utility requires the `JAVA_HOME` environment variable to be set. If you do not have Java installed, you can optionally use the Java version provided with the product and available in:

IBM® Workload Scheduler

```
<INST_DIR>/TWS/JavaExt/jre/jre
```

Dynamic Workload Console

```
<DWC_INST_DIR>/java/jre/bin
```

Streamlogon user

The streamlogon user of any job run in the IBM Workload Scheduler environment (jobs running on Windows® only)

For all other users of IBM Workload Scheduler, no action is required if their passwords change.



Note: After changing any password, restart WebSphere Application Server Liberty Base.

Before changing any passwords, you must first change the password at the operating system level using native commands, as follows:

On UNIX operating systems

use the `passwd` command.

On Windows operating systems

use the `net user` command.

If you use special characters in the password, ensure you use a "\" (backslash) before the special character. The following rules apply:

On Windows™ operating systems:

Passwords for users can include any alphanumeric characters and `()!?=^*/~[]$+_!@`-#`.

On UNIX™ and LINUX systems:

Passwords for users can include any alphanumeric characters and `()!?=*_~+.-`.

See [Table 82: If and where password changes are required on page 413](#) to determine if a change of password requires actions to be taken for a role on the different IBM Workload Scheduler components. Look up the role and the component and determine from the corresponding table cell where the changes must be made:

- If the cell contains a "✓", make the change on the system where the indicated component is running
- If the cell contains "MDM", make the change on the master domain manager to which the component belongs

Table 82. If and where password changes are required

Role	MDM	BKM	FTA	FTA + CONN
IBM Workload Scheduler instance owner (Windows®)	✓	✓	✓	✓
WebSphere Application Server Liberty Base user	✓	✓		✓
Database user	✓	✓		
Streamlogon user (Windows®)	✓	✓	MDM	MDM

For example, if you are the `<TWS_user>` (the instance owner) of a fault-tolerant agent, you need to implement the password change on the system where the fault-tolerant agent is installed, but if you are also the streamlogon user of jobs running on that system, the changes required for the new password must be applied at the master domain manager to which the fault-tolerant agent belongs.

Changing the WebSphere Application Server Liberty Base user ID and password

Procedure to modify the WebSphere Application Server Liberty Base user ID and password.

To modify the WebSphere Application Server Liberty Base user ID and password, you must update the `wauser_variables.xml` configuration file. If you are modifying the password for the primary WebSphere Application Server Liberty Base administrator, then you must also update the `useropts` file.

1. Browse to the `wauser_variables.xml` template, which is located in:

Dynamic Workload Console

DWC_home

```
/usr/servers/dwcServer/configDropins/templates
```

master domain manager

TWA_home

```
/usr/servers/engineServer/configDropins/templates
```

Dynamic Workload Console

```
DWC_home\usr\servers\dwcServer\configDropins\templates
```

master domain manager

```
TWA_home\usr\servers\engineServer\configDropins\templates
```

2. Copy the `wauser_variables.xml` template to a temporary directory.
3. Edit the file as necessary, specifying the new password.

The contents of the `wauser_variables.xml` file is as follows:

```
<server description="wauser_var">
  <variable name="user.twsuser.id" value="\$(wlpUser)"/>
  <variable name="user.twsuser.password" value="\$(wlpPassword)"/>
</server>
```

where:

user.twsuser.id

Is the WebSphere Application Server Liberty Base user ID.

user.twsuser.password

Is the new WebSphere Application Server Liberty Base password.

4. Optionally, create a backup copy of the configuration file in a different directory, if the file is already present.
5. Copy the `wauser_variables.xml` file to the `overrides` directory. Changes are effective immediately.

Dynamic Workload Console

```
DWC_DATA_dir/usr/servers/dwcServer/configDropins/overrides
```

master domain manager

```
TWA_DATA_DIR/usr/servers/engineServer/configDropins/overrides
```

Dynamic Workload Console

```
DWC_home\usr\servers\dwcServer\configDropins\overrides
```

master domain manager

```
TWA_home\usr\servers\engineServer\configDropins\overrides
```

6. Update the USERNAME and PASSWORD in the `useropts` file on the following workstations:
 - on every command-line client that points to your workstation.
 - on every fault-tolerant agent in your environment that has an HTTP/HTTPS connection defined in `localopts` that points to your workstation. The HTTP/HTTPS connection is used to submit a predefined job or job stream.
 - in the engine connection parameters on every connected Dynamic Workload Console.

Change password used by command-line clients to access the master domain manager

About this task

If you have changed the password of the WebSphere Application Server Liberty Base user that command-line clients use to connect to the master domain manager, the connection parameters must be updated.

Follow this procedure:

1. Identify all systems that have a command line client remote connection defined with the master domain manager
2. On these workstations, open the user options files (one for each user). The default file name is `User_home/.TWS/useropts`, but if you have more than one instance of IBM Workload Scheduler on a system, you might have implemented separate user options files to make separate connections, in which case consult the `useropts` key in the `localopts` file on each instance to determine the name of the specific `useropts` file for that instance.
3. For each file, locate the password key (encrypted) and change its value to that of the new password in plain text, enclosed in double quotation marks. The password is saved in clear, but will be encrypted at first logon of the User ID.
4. Save the files.
5. Check if the following file exists: `Root_home/.TWS/useropts`. If it does, change the password in the same way.

Change password used by fault-tolerant agent systems to access the master domain manager (for conman)

About this task

If you have changed the password of the WebSphere Application Server Liberty Base user that is used by fault-tolerant agents with an HTTP or HTTPS connection defined in the local options that points to the master domain manager, the connection parameters must be updated.

Follow this procedure:

1. Identify all fault-tolerant agents with an HTTP or HTTPS connection defined in the local options that points to the master domain manager.
2. On these workstations, open the user options file `<Root_home>/ .TWS /useropts`
3. Locate the password key (encrypted) and change its value to that of the new password in plain text, enclosed in double quotation marks. The password is saved in clear, but will be encrypted at first logon of the User ID.
4. Save the file.

Update the engine connection parameters in the GUIs

About this task

If you have changed the password of the WebSphere Application Server Liberty Base user that is used by the Dynamic Workload Console to connect to the distributed engine, the engine connection parameters must be updated, as follows:

1. On each instance of the Dynamic Workload Console locate the page where you modify the distributed engine connection parameters
2. Change the password and submit the page.

Windows™ - update Windows™ services

About this task

On Windows™, the `TWS_user` account is used to start the following services:

- IBM Token Service for `TWS_user`
- IBM Workload Scheduler for `TWS_user`

The password must be updated in the properties of these services, or they are not able to start at next reboot. This is done as follows:

1. Stop all IBM Workload Scheduler processes. See [Unlinking and stopping IBM Workload Scheduler on page 417](#) for details.
2. Restart all IBM Workload Scheduler processes using the StartUp command.

Change the IBM Workload Scheduler user definition

About this task

If the user ID is used within IBM Workload Scheduler to run jobs, follow this procedure:

1. Run the composer modify user command. The user details of the selected user are written to a temporary file, which is opened.
2. Edit the password field so that it contains the new password value delimited by double quotation marks characters (").

3. Save the file, and the contents are added to the database.
4. To make the change immediately effective in the current plan, issue the `conman altpass` command.

For the full syntax of these commands see the *User's Guide and Reference*.

Unlinking and stopping IBM Workload Scheduler

About this task

Before you perform an upgrade or uninstall, install a fix pack, or perform maintenance activities, ensure that all IBM Workload Scheduler processes and services are stopped. Follow these steps:

1. If you have jobs that are currently running on the workstation, wait for them to finish. To determine which are not finished, check for jobs that are in the `exec` state. When there are no jobs in this state, and you have allowed sufficient time for all events to be distributed in your network, you can continue with the rest of the procedure.
2. If the workstation that you want to stop is not the master domain manager, unlink the workstation by issuing the following command from the command line of the master domain manager:

```
conman "unlink workstationname;noask"
```

3. Stop WebSphere Application Server Liberty Base by using the `conman stopappserver` command (see [Starting and stopping the application server and appservman on page 436](#)).
4. All IBM Workload Scheduler processes on the workstation must then be stopped manually. From the command line, while logged on as the `<TWS_user>`, enter the following command:

```
conman "stop;wait"
```

5. From the command line, stop the netman process as follows:

UNIX®

Run the `conman "shut"` command.



Note: Do not use the UNIX® kill command to stop IBM Workload Scheduler processes.

Windows®

From the IBM Workload Scheduler home directory, run the `shutdown.cmd` command.

6. To stop dynamic agents, run the `ShutDownLwa` command.
7. To stop the SSM agent, perform the following steps:
 - On Windows®, stop the service IBM Workload Scheduler SSM Agent (for `<<TWS_user>>`).
 - On UNIX®, run the `stopmon` command.

What to do next

To verify if there are services and processes still running:

UNIX®

Enter the command: `ps -u <TWS_user>` Verify that the following processes are not running: netman, mailman, batchman, writer, jobman, JOBMAN, stageman, logman, planman, monman, ssmagent.bin, and appservman.

Windows®

Run **Task Manager**, and verify that the following processes are not running: netman, mailman, batchman, writer, jobman, stageman, JOBMON, tokensrv, batchup, logman, planman, monman, ssmagent, and appservman.

Also, ensure that no system programs are accessing the directory or its sub-directories, including the command prompt and Windows® Explorer.

Changing the properties for the database

About this task

When you installed IBM Workload Scheduler, you supplied the database name, the server name, port, the host name of the database server, and other information.

If you need to change any of the database properties such as host name, port, or database name, use the `datasource_<db_vendor>.xml` template file to reflect these changes in the application server on the master domain manager or on the Dynamic Workload Console.

If you want to change any of these properties, perform the following steps:

1. Change the configuration of the IBM Workload Scheduler application server so that it points correctly to the changed database configuration, as follows:
 - a. Browse to the following path:

Dynamic Workload Console

```
DWC_DATA_dir/usr/servers/dwcServer/configDropins/templates/datasources
```

master domain manager

```
TWA_DATA_DIR/usr/servers/engineServer/configDropins/templates/datasources
```

Dynamic Workload Console

```
DWC_home\usr\servers\dwcServer\configDropins\templates\datasources
```

master domain manager

```
TWA_home\usr\servers\engineServer\configDropins\templates\datasources
```

- b. Copy the `datasource_<db_vendor>.xml` file to a temporary location.
- c. Modify the following properties in the file based on the values you changed in your database configuration:

db.serverName

The name or IP address of the database server

db.portNumber

The port number of the database server

db.databaseName

The database name

db.user

The database user

db.password

The database user password. You can optionally encrypt the password, as described in the topic about encrypting passwords in *IBM Workload Scheduler: Planning and Installation*.

db.driver.path

The path to the JDBC drivers. IBM® Workload Scheduler is supplied using the JDBC driver type 4 for DB2® and type 2 for Oracle. However, each can use the other driver type, if necessary. IBM® Workload Scheduler Software Support might ask you to change to this driver. This procedure must only be performed under the control of IBM® Workload Scheduler Software Support.

db.sslConnection

The setting for the SSL connection. `true` indicates that SSL connection is enabled, `false` indicates that SSL connection is disabled. .

- d. Browse to the following path:

Dynamic Workload Console

```
DWC_DATA_dir/usr/servers/dwcServer/configDropins/overrides
```

master domain manager

```
TWA_DATA_DIR/usr/servers/engineServer/configDropins/overrides
```

Dynamic Workload Console

```
DWC_home\usr\servers\dwcServer\configDropins\overrides
```

master domain manager

```
TWA_home\usr\servers\engineServer\configDropins\overrides
```

- e. Create a backup of the `datasource_<db_vendor>.xml`.
- f. Replace the `datasource_<db_vendor>.xml` file with the file you edited. The changes are effective immediately.

- g. On the master domain manager only, edit the file `CLIconfig.properties`, in the path `TWA_DATA_DIR/broker/config`, by updating the value for the property `com.ibm.tdwb.dao.rdbms.jdbcPath` to reflect the JDBC URL specified. The following is an example of a JDBC value:

```
<database_type</varname>>://<<varname>hostname>:<port>/<dbName>
```

- h. On the master domain manager only, edit the file `DAOCommon.properties`, in the path `TWA_DATA_DIR/broker/config`, as follows:

- In the first line specify the **rdbmsName** for all DBs.
- For Oracle only, change all lines and specify the `TWS_Oracle_User`.

- i. On the master domain manager only, edit the file `TWSConfig.properties`, in the path `TWA_DATA_DIR/usr/servers/engineServer/resources/properties/TWSConfig.properties`. If you are using Oracle, Informix, and MSSQL change the first line. For Oracle only, change all lines. Consider the following example:

```
com.ibm.tws.dao.rdbms.rdbmsName = ORACLE
com.ibm.tws.dao.rdbms.modelSchema = <TWS_Oracle_User>
com.ibm.tws.dao.rdbms.eventRuleSchema = <TWS_Oracle_User>
com.ibm.tws.dao.rdbms.logSchema = <TWS_Oracle_User>
```

Changing the workstation host name or IP address

When you change the host name, the IP address or both on the workstations of your IBM Workload Scheduler environment to have it function properly, you must report the changed value on:

- The WebSphere Application Server Liberty Base if the following components changed the host name, the IP address or both:
 - Master domain manager
 - Backup master domain manager
 - Connector or Z connector
 - Dynamic Workload Console

For more information, see [Reporting the changes in the WebSphere Application Server Liberty Base configuration file on page 421](#).

- The following components if the workstation where you installed the RDBMS changed the host name, the IP address or both:
 - Master domain manager
 - Backup master domain manager
 - Dynamic domain manager
 - Backup dynamic domain manager

For more information, see [Changing the properties for the database on page 418](#).

- The workstation definitions if you installed the following components:
 - Master domain manager
 - Backup master domain manager
 - Dynamic domain manager
 - Backup dynamic domain manager

- Fault-tolerant agent and standard agent
- Domain manager

For more information, see [Reporting the changed host name or IP address in the workstation definition on page 422](#).

Reporting the changes in the WebSphere Application Server Liberty Baseconfiguration file

About this task

If the host name or IP address is changed for the following components, then you must report the changed value in the WebSphere Application Server Liberty Base `host_variables.xml` configuration file:

- Master domain manager
- Backup master domain manager
- Dynamic domain manager
- Backup dynamic domain manager
- Dynamic Workload Console

1. Stop the WebSphere Application Server Liberty Base.
2. Obtain the changed host name, IP address, or both.
3. Verify that the value for the properties listed below were changed with the actual values:
 - Old Hostname
 - New Hostname
 - The host names for the specific port properties

If these values are different from the actual host name or IP address values, proceed with [Step 4 on page 421](#). If these values are not changed, skip the steps below.

4. Create a back up of and then modify the values in the `host_variables.xml` file located in

On UNIX operating systems

```
TWA_DATA_DIR/usr/servers/engineServer/configDropins/overrides
```

On Windows operating systems

```
TWA_home\usr\servers\engineServer\configDropins\overrides
```

5. Propagate the changes to the interfaces as follows:

Address of the master domain manager changes

- On each fault-tolerant agent, dynamic agent, and standard agent you configured to connect to the **conman** command line, update the **host** parameter present in the "**Attributes for CLI connections**" section in the `localopts` file. Usually you have the **host** parameter defined in the `localopts` file of the workstations you use to submit predefined jobs and job streams (sbj and sbs commands).
- On every command-line client, update the **host** parameter present in the "**Attributes for CLI connections**" section in the `localopts` file.

- On the Dynamic Workload Console, update the engine connections.
- On all of the master components:
 - a. Run JnextPlan.
 - b. Update the value for the **Master.Switch.HostName** parameter in the `SwitchBroker.properties` file located in `TWA_DATA_DIR/broker/config/SwitchBroker.properties`.
 - c. From the broker command line, run `exportserverdata` to extract a list of URIs containing the old hostname to a text file, then run `importserverdata` providing the updated file in input to update the changed host name. For more information about the commands, see the section about using utility commands in the dynamic environment in *User's Guide and Reference*.

Address of the Dynamic Workload Console changes

Notify all the users of the new web address.

Reporting the changed host name or IP address in the workstation definition

About this task

Run this procedure if you changed the host name or IP address on the following components:

- Master domain manager
- Backup master domain manager
- Fault-tolerant agent and standard agent
- Domain manager

To modify the host name or the IP address on the workstation definition, perform the following steps:

1. Use **composer** or the Dynamic Workload Console to check the workstation definition stored in the database for the IBM Workload Scheduler instance installed on the workstation where the IP address or the host name changed.
2. Verify the **node** attribute contains the new host name or IP address. If this value is changed proceed with Step 3 on [page 422](#). If this value is not changed skip the steps below.
3. Change the value of the **node** parameter with the new value.
4. Refresh the new workstation definition into the plan. Do it immediately if you are changing the host name or the IP address of a master domain manager or a domain manager. If you are changing them on a workstation that is not a master domain manager or a domain manager you can wait the next scheduled plan generation to refresh your workstation definition in the Symphony file. In this case during this production day you cannot run jobs on this workstation. To generate the plan, perform the following steps:
 - a. Run **optman ls** and take note of the actual value of the **enCarryForward** parameter.
 - b. If this value is not set to **all**, run


```
optman chg cf=ALL
```

to set it to **all**
 - c. Add the new workstation definition to the plan, by running:

```
JnextPlan -for 0000
```

- d. Reassign the original value to the **enCarryForward** parameter.

Reporting the changed host name or IP address of the dynamic workload broker server

About this task

The dynamic workload broker server is a component that IBM Workload Scheduler installs when you install the following components:

- Master domain manager
- Backup master domain manager
- Dynamic domain manager
- Backup dynamic domain manager

If you changed the host name or the IP address on the dynamic workload broker server, or if you installed a new one run the procedure described in [Reporting the changes in the WebSphere Application Server Liberty Baseconfiguration file on page 421](#).

If you changed the host name or the IP address on a master domain manager or backup master domain manager and you ran the [Reporting the changes in the WebSphere Application Server Liberty Baseconfiguration file on page 421](#) procedure, skip this section.

If you changed the host name or the IP address on the dynamic domain manager or backup dynamic domain manager you do not need to change the definition of your broker workstation (type **broker**), because the value of the **node** attribute is set to the *localhost* value to allow to switch between the dynamic workload broker server and its backup.

After you ran the procedure, propagate the changes to the dynamic agent and update the **ResourceAdvisorURL** property in the `JobManager.ini` file on each agent connected to that dynamic workload broker server, by performing the following steps:

1. Run the following command to stop the agent:

```
ShutDownLwa
```

2. Edit the `JobManager.ini` file and change the host name or the IP address in the **ResourceAdvisorURL** property.
3. Run the following command to start the agent:

```
StartUpLwa
```

Perform the following changes:

1. Open the `JobDispatcherConfig.properties` file and change the value of the **JDURL=https://host_name** property to reflect the new host name or IP address.
2. Open the `CliConfig.properties` file and change the value of the **ITDWBServerHost=/host_name** property to reflect the new host name or IP address.

3. Open the `ResourceAdvisorConfig.properties` file and change the value of the **ResourceAdvisorURL=https://*host_name*** property to reflect the new host name or IP address.
4. From the `<TWA_home>/TDWB/bin` directory, run the following command:

On Windows operating systems:

```
exportserverdata.bat
```

On UNIX and Linux operating systems:

```
exportserverdata.sh
```

This command extracts a list of URIs (Uniform Resource Identifier) of all the dynamic workload broker instances from the IBM Workload Scheduler database and copies them to a temporary file. By default, the list of URIs is saved to the `server.properties` file, located in the current directory.

5. Change all the entries that contain the old host name to reflect the new host name.
6. Place the file back in the database, by running the following command:

On Windows operating systems:

```
importserverdata.bat
```

On UNIX and Linux operating systems:

```
importserverdata.sh
```

7. Stop WebSphere Application Server Liberty Base, by running the following command:

```
stopAppServer
```

8. Start WebSphere Application Server Liberty Base, by running the following command:

```
startAppServer
```

Reporting the changed host name or IP address of the dynamic agent

About this task

If you changed the host name or the IP address on the workstation where you installed the dynamic agent, the changes are automatically reported stopping and starting the agent using the following commands:

1. To stop the agent:

```
ShutDownLwa
```

2. To start the agent:

```
StartupLwa
```



Note: Do not modify manually the value of the **node** parameter in the dynamic agent workstation definition.

Changing the security settings

This section describes how to modify the security settings of IBM Workload Scheduler.

About this task

A number of template files are available for customizing various security settings on the application server:

ssl_variables.xml

auth_basicRegistry_config.xml

File-based authentication

auth_IDS_config.xml

IBM® Directory Server

auth_OpenLDAP_config.xml

OpenLDAP

auth_OpenLDAP_config.xml

Windows Server Active Directory

For the procedure to customize the security settings, see the section about configuring a common LDAP for both the master and the console in *Planning and Installation Guide*.

For the settings related to SSL, see Setting connection security. For the settings related to the passwords of the database access users, see [Changing key IBM Workload Scheduler passwords on page 412](#). You can also change other settings, such as the active user registry or the local operating system ID and password.

- For more information about the procedure to make any changes to the WebSphere Application Server Liberty Base properties, see [Configuring IBM Workload Scheduler using templates on page 428](#).
- To determine which properties are to be changed, see:
 - [Configuring authentication on page 252](#), for information about the properties to be changed to modify your user registry configuration for user authentication.
 - see the scenario about the connection between the Dynamic Workload Console and the IBM Workload Scheduler components in *Planning and Installation Guide*, for information about the properties to be changed to configure SSL communication between the different interfaces and the IBM Workload Scheduler engine.
 - [Modifying your RDBMS server on page 363](#), for information about the procedure to be performed when modifying your RDBMS server.
 - [Changing the properties for the database on page 418](#), for information about changing the database properties such as database name, the server name, port, the host name of the database server.
 - [Changing key IBM Workload Scheduler passwords on page 412](#), for information about how to use the properties to determine the procedure required for changing key passwords.
- To change the text file of the current security properties, perform the following steps:
 1. Edit the text file and locate the properties you need to change.
 2. Make any required changes to the properties.

Do not change any other properties.

Managing the event processor

About this task

The only maintenance issue for the event processor is the management of the EIF event queue, `cache.dat`. The event queue is circular, with events being added at the end and removed from the beginning. However, if there is no room to write an event at the end of the queue it is written at the beginning, overwriting the event at the beginning of the queue.

To increase the size of the event processor queue, follow this procedure:

1. At the workstation running the event processor, browse to the following path:

On Windows operating systems

```
<TWA_home>\TWS\stdlist\appserver\engineServer\temp\TWS\EIFListener
```

On UNIX operating systems

```
<TWA_DATA_DIR>/stdlist/appserver/engineServer/temp/TWS/EIFListener
```

2. Edit the `EIFListener` file and locate the keyword:

```
BufEvtMaxSize
```

3. Increase the value of this keyword, according to your requirements.
4. Stop and restart WebSphere Application Server Liberty Base using the `stopappserver` and `startappserver` commands (see [Starting and stopping the application server and appservman on page 436](#)).

Automatically initializing IBM Workload Scheduler instances

On UNIX systems, you can automatically initialize IBM Workload Scheduler instances during operating system startup.

About this task

On UNIX™ systems, you can ensure that your IBM Workload Scheduler instances are automatically initialized during operating system startup.

For AIX®, Solaris, HP-UX and some Linux™ operating systems that use a traditional **init** like System V, you can do this by adding an IBM Workload Scheduler service to the **init** process of your operating system. Use the sample start script `iwa_init_<installation user>` located in `TWA_home/TWS/config` and add it to the appropriate run level after customizing it as necessary.

For some Linux™ distributions that use `systemd` as the default initialization system, such as RedHat Enterprise Linux™ v7.0 and SUSE Linux™ Enterprise Server V12, a sample service file, `iwa.service`, is provided located in the path `TWA_home/TWS/config` that is already configured to support the automatic initialization of IBM Workload Scheduler instances at startup.

Perform the following steps:

For UNIX operating systems that use the traditional init system such as System V :

Configure the script and then register the service.

1. Create a copy of the `iwa_init_<installation user>` script based on your requirements. Provide the following information, depending on the operating system you use:

Required-Start

On Linux™ systems, specify the precondition services

Default-Start

On Linux™ systems, specify the required runlevels. For example, specify runlevels 2, 3, and 5.

2. Browse to the appropriate system-dependent folder, as follows:

Supported Linux™ operating systems

Save the script in the `/etc/init.d` folder and register the service using the `insserv -v script_name` command.

Supported AIX® operating systems

Save the script to the appropriate `rcrunlevel.d` folder. Rename the script according to the runlevel script definition. For example, `Ssequence_numberservice_name`, as in `S10iwa_init_<installation user>`.

Supported Solaris operating systems

Save the script in the `/etc/init.d` folder and link the script to an appropriate file in the `rcrunlevel.d` folder. For example, `Ssequence_numberservice_name`, as in `S10tws860ma`.

Supported HP-UX operating systems

Save the script in the `/sbin/init.d` folder and link the script to an appropriate file in the `rcrunlevel.d` folder. For example, `Ssequence_numberservice_name`, as in `S10tws860ma`.



Note: If you run this script on a master domain manager or on a backup master domain manager, the script has no effect on the database.

For more information about the `inittab`, `init.d`, `insserv`, and `init` commands, see the reference documentation for your operating system.

For Linux™ distributions that use systemd as the default initialization system:

This procedure uses the `iwa.service` sample service that is already customized to automatically initialize IBM Workload Scheduler instances at system startup.

1. Copy the sample service provided, `iwa.service`, located in the path `TWA_home/TWS/config` to the following path on the Linux™ system `/etc/systemd/system/`
2. To make systemd aware of the service, invoke the `systemctl daemon-reload` command.
3. Start the service submitting the following command:

```
systemctl start iwa.service
```

4. Verify the status by submitting the following command:

```
systemctl status iwa.service
```

5. Stop the service by submitting the following command:

```
systemctl stop iwa.service
```

6. Finally, enable the service so that it is activated by default when the system boots by submitting the following command:

```
systemctl enable iwa.service
```

Configuring IBM Workload Scheduler using templates

Starting from version 9.5, IBM Workload Scheduler has moved from WebSphere® Application Server to WebSphere Application Server Liberty Base. As a result, the utilities known as wastools have been replaced with a number of templates addressing widely used configurations. You can now configure WebSphere Application Server Liberty Base to work with IBM Workload Scheduler using the templates provided or define your custom `.xml` files containing your own configuration settings. Templates are available for both the master domain manager and the Dynamic Workload Console.

See [Table 83: Correspondence between wastools and templates on page 429](#) to find the mapping between wastools and templates. The table indicates both the file containing the current configuration in use, as well as the template files available to modify the current configuration.

Templates for the master domain manager are stored in the following paths:

On UNIX operating systems

```
TWA_home/usr/servers/engineServer/configDropins/templates
```

On Windows operating systems

```
TWA_home\usr\servers\engineServer\configDropins\templates
```

Templates for the Dynamic Workload Console are stored in the following paths:

On UNIX operating systems

```
DWC_home/usr/servers/dwcServer/configDropins/templates
```

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:

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
```



Note: Do not edit the files in the `templates` directory because they will be overwritten when upgrading to new version or fix pack.

To configure WebSphere Application Server Liberty Base to work with IBM Workload Scheduler, use the template files provided in the `templates` folder or create your custom `.xml` files containing your configuration settings. WebSphere Application Server Liberty Base retrieves the `.xml` files from the `overrides` folder and applies the configuration settings defined in each file. The file name is irrelevant, because WebSphere Application Server Liberty Base analyzes each `.xml` file for its contents.

The template files provided refer to commonly used configurations. If you want to implement different configurations, for example a custom authorization mechanism, you can create a custom `.xml` file, containing your configuration settings in this section:

```
<server description="My custom configuration description">

</server>
```

Ensure you remove any obsolete `.xml` files, to prevent WebSphere Application Server Liberty Base from parsing unwanted files.

If you use the provided templates, ensure you follow this procedure:

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.

Table 83. Correspondence between wastools and templates

Function	Current configuration file in use	Template file for customization	wastool
Datasource settings	<code>overrides/datasource.xml</code>	<code>templates/datasources/datasource_vendor.xml</code>	<code>changeDataSourceProperties</code> <code>showDataSourceProperties</code>

Table 83. Correspondence between wastools and templates (continued)

Function	Current configuration file in use	Template file for customization	wastool
Hostname and port settings	<ul style="list-style-type: none"> overrides/host_variables.xml overrides/ports_variables.xml 	<ul style="list-style-type: none"> Not applicable. The <code>host_variables.xml</code> file is very simple and therefore is located only in the <code>overrides</code> folder. templates/ports_variables.xml 	<p>changeHostProperties</p> <p>showHostProperties</p>
Authentication settings	overrides/wauser_variables.xml	Not applicable. The <code>wauser_variables.xml</code> file is very simple and therefore is located only in the <code>overrides</code> folder.	changePassword
Trace settings	Traces are disabled by default, so no file is present in the <code>overrides</code> folder. Copy the <code>trace.xml</code> file to the <code>overrides</code> folder to enable traces.	templates/trace.xml	changeTraceProperties
z/OS engine settings for the Dynamic Workload Console	overrides/connectionFactory.xml	zconnector/connectionFactory.xml	createZosEngine (Dynamic Workload Console installation only)
SSL connections and certificates	<ul style="list-style-type: none"> overrides/authentication_config.xml overrides/ssl_variables.xml 	<p>File-based:</p> <p>authentication/auth_basicRegistry_config.xml</p> <p>IBM® Directory Server:</p> <p>authentication/auth_IDS_config.xml</p>	<p>showSecurityProperties</p> <p>changeSecurityProperties</p>

Table 83. Correspondence between wastools and templates (continued)

Function	Current configuration file in use	Template file for customization	wastool
		<p>OpenLDAP:</p> <p>authentication/ldap_authentication.xml</p> <p>Microsoft Server Active Directory:</p> <p>authentication/ad_authentication_config.xml</p> <p>OpenID:</p> <p>authentication/openid_connect.xml</p> <p>ssl_variables.xml</p>	

Templates are divided into the following directories:

defaults (files used by the installation process)

- ports_config.xml
- ports_variables.xml
- ssl_config.xml
- ssl_variables.xml

overrides (configuration files)

- authentication_config.xml
- connectionFactory.xml (Dynamic Workload Console installation only)
- datasource.xml
- host_variables.xml
- jvm.options
- ports_variables.xml
- ssl_variables.xml
- wauser_variables.xml

templates (templates available for customization)

authentication

- `auth_AD_config.xml`
- `auth_basicRegistry_config.xml`
- `auth_IDS_config.xml`
- `auth_OpenLDAP_config.xml`
- `openid_connect.xml`

datasources

- `datasource_db2.xml`
- `datasource_derby.xml` (Dynamic Workload Console installation only)
- `datasource_ids.xml` (also for oneDB)
- `datasource_mssql.xml`
- `datasource_oracle.xml`

zconnectors (Dynamic Workload Console installation only)

- `connectionFactory.xml`
- `ports_variables.xml`
- `ssl_variables.xml`
- `trace.xml`

For information about how to change database properties, see [Changing the properties for the database on page 418](#).

For information about authentication settings, see [Configuring authentication on page 252](#).

For information about configuring a z/OS engine, see the section about defining az/OS engine in the Z connector in *IBM Z Workload Scheduler: Planning and Installation*.

For information about logs and traces, see the section about logging and tracing in *Troubleshooting Guide*.

WebSphere Application Server Liberty Base tasks

The following WebSphere Application Server Liberty Base tasks might need to be performed:

Application server - starting and stopping

Use the `startappserver` and `stopappserver` commands or the equivalent from the Dynamic Workload Console to start or stop the WebSphere Application Server Liberty Base. For a description of these commands, see *IBM Workload Scheduler: User's Guide and Reference*.

These commands also stop appservman, the service that monitors and optionally restarts WebSphere Application Server Liberty Base.

If you do not want to stop appservman, you can issue startAppServer or stopAppServer, supplying the `-direct` argument. These scripts are located in `TWA_home/appservertools`.

The complete syntax of startAppServer and stopAppServer is as follows:

UNIX™

Start the application server

```
./startAppServer.sh [-direct]
```

Stop the application server

```
./stopAppServer.sh [-direct]
```



Note: If your WebSphere Application Server Liberty Base is for the Dynamic Workload Console, you must use the following syntax:

```
./stopAppServer.sh [-direct]
                    [-user <user_ID>]
                    -password <password>]
```

The user ID and password are optional only if you have specified them in the `soap.client.props` file located in the properties directory of the WebSphere Application Server Liberty Base profile.

Unlike the master domain manager installation, when you install the Dynamic Workload Console the `soap.client.props` file is not automatically customized with these credentials.

Windows™

Start the application server

```
startAppServer.bat [-direct]
```

Stop the application server

```
stopAppServer.bat [-direct]
                  [-wlpHome <installation_directory>]
                  [-options <parameters>]]
```

z/OS

Start the application server

```
./startAppServer.sh [-direct]
```

Stop the application server

```
./stopAppServer.sh [-direct]
```

where the arguments are as follows:

-direct

Optionally starts or stops the application server without starting or stopping the application server monitor appservman.

For example, you might use this after changing some configuration parameters. By stopping WebSphere Application Server Liberty Base without stopping appservman, the latter will immediately restart WebSphere Application Server Liberty Base, using the new configuration properties.

This argument is mandatory on UNIX™ when the product components are not integrated.

-options *parameters*

Optionally supplies parameters to the WebSphere Application Server Liberty Base startServer or stopServer commands. See the WebSphere Application Server Liberty Base documentation for details.

-wlpHome *installation_directory*

Defines the WebSphere Application Server Liberty Base installation directory, if it is not the default value.

Application server - automatic restart after failure

If you experience any problems with the application server failing, a service is available that not only monitors its status, but can also restart it automatically in the event of failure. The service is called appservman, and it is enabled and controlled by the local options on the computer where the application server is running.

The following sections describe the service, how it works, and how it is controlled:

- [Appservman - how it works on page 434](#)
- [Controlling appservman on page 435](#)
- [Starting and stopping the application server and appservman on page 436](#)
- [Monitoring the application server status on page 436](#)
- [Obtaining information about application server failures on page 437](#)
- [Events created by appservman on page 437](#)

Appservman - how it works

Appservman is a service that starts, stops and monitors the application server. It also optionally restarts the application server in the event that the latter fails. Appservman can be controlled not just from nodes running the application server, but also from any other node running conman.

It is launched as a service by netman when starting IBM Workload Scheduler, and it itself then launches the application server. Netman also launches it when the conman startappserver command is run.

Appservman is stopped when IBM Workload Scheduler is shut down. In addition, Netman stops both the application server and appservman when you use the conman stopappserver command, or, on Windows™ only, when you issue the Shutdown – appsvr command.

While it is running `appservman` monitors the availability of the application server, sending events that report the status of the application server. If the automatic restart facility is enabled, and the application server fails, the service determines from the restart policy indicated in the `localopts` options if it is to restart the application server. If the policy permits, it will restart the application server, and send events to report its actions.

Using the `startappserver` and `stopappserver` commands to stop the application server and `appservman`.

Controlling `appservman`

`Appservman` is controlled by the following local options (in the `localopts` file):

Appserver auto restart

Determines if the automatic restart facility is enabled.

The default is *yes*. To disable the option set it to *no*.

Appserver check interval

Determines how frequently the service checks on the status of the application server. You should not set this value to less than the typical time it takes to start the application server on the computer.

The default is every 3 minutes.

Appserver min restart time

Determines the minimum time that must elapse between failures of the application server for the automatic restart to work. This option stops `appservman` from immediately restarting the application server if it fails on initial startup or when being restarted.

The default is 2 minutes.

Appserver max restarts

Determines the maximum number of times that `appservman` will automatically restart the application server within a time frame determined by you (`Appserver count reset interval`).

The default is 5 restarts.

Appserver count reset interval

Determines the time frame for the maximum number of restarts (`Appserver max restarts`).

The default is 24 hours.

How to use the options

The default settings are a good starting point. Follow the indications below if you are not satisfied that the settings are maintaining the correct availability of the application server:

- If the application server is not restarting after failure, check the following:
 - That the Appserver auto restart is set to `yes`.
 - That the Appserver check interval is not set to too high a value. For example, if this value is set to 50 minutes, instead of the default 3, an early failure of the application server might wait 45 minutes before being restarted.
 - That Appserver min restart time is sufficient for the application server to fully restart. If, when the server checks the status of the application server, it finds that the application server is still starting up, in some circumstances it is not able to distinguish the starting-up state from the failed state, will report it as failed, and try and restart it again. With the same result. This will continue until Appserver max restarts is exceeded. If this is the case, make Appserver min restart time larger.
- If the application server is failing infrequently, but after several failures is not restarting, set the Appserver max restarts option to a higher value or the Appserver count reset interval to a lower value, or both. In this case it might be advantageous to study the pattern of failures and tailor these options to give you the required availability

Starting and stopping the application server and appservman

If you need to stop and restart WebSphere Application Server Liberty Base, for example to implement a change in the application server configuration, use the following conman commands:

conman stopappserver[*domain!*]workstation [;wait]

This command stops the application server and appservman. You can optionally stop the application server on a remote workstation. The optional `;wait` parameter tells conman to suspend processing until the command reports that both the application server and the service have stopped.

conman startappserver[*domain!*]workstation [;wait]

This command starts the application server and appservman. You can optionally start the application server on a remote workstation. The optional `;wait` parameter tells conman to suspend processing until the command reports that both the application server and the service are up and running.

To stop and start the application server without stopping appservman, see [Application server - starting and stopping on page 432](#).

Monitoring the application server status

To see the current status of the application server at any time view the STATE field in the workstation details.

This field contains a string of characters that provide information about the statuses of objects and processes on the workstation. The state of the application server is a one-character flag in this string, which has one of the following values if the application server is installed:

[A|R]

where:

A

WebSphere Application Server Liberty Base is running.

R

WebSphere Application Server Liberty Base is restarting.

If WebSphere Application Server Liberty Base is down or if it was not installed, neither value of the flag is present in the STATE entry.

Obtaining information about application server failures

Appservman does not provide information about why the application server has failed. To obtain this information, look in the application server log files (see the *IBM Workload Scheduler: Troubleshooting Guide*).

Events created by appservman

Appservman sends an event called *ApplicationServerStatusChanged* from the `TWSObjectsMonitor` provider to the configured event monitoring process every time the status of the application server changes.

Application server - encrypting the profile properties files

For more information about encrypting passwords in WebSphere Application Server Liberty Base profile properties files, see the related documentation, for example [securityUtility command](#).

This utility requires the JAVA_HOME environment variable to be set. If you do not have Java installed, you can optionally use the Java version provided with the product and available in:

IBM® Workload Scheduler

```
<INST_DIR>/TWS/JavaExt/jre/jre
```

Dynamic Workload Console

```
<DWC_INST_DIR>/java/jre/bin
```

Application server - configuration files backup and restore

On UNIX operating systems, all configuration files are stored in the `<TWA_DATA_DIR>/usr` and `<DWC_DATA_dir>/usr`.

On Windows operating systems, backup the files in `<TWA_home>\usr` and `<DWC_home>\usr`.

Application server - changing the host name or TCP/IP ports

To modify the host name of the computer where the application server is installed, or the TCP/IP ports it uses, use the following templates:

host_variables.xml

Specify the hostname in the `variable name` property.

ports_variable.xml

Edit the following properties as required:

- host.http.port
- host.https.port
- host.bootstrap.port
- host.bootstrap.port.sec

To disable a port, set its value to -1.

Edit the files as required using the following basic procedure:

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 more information about the procedure, see [Configuring IBM Workload Scheduler using templates on page 428](#). For more information about WebSphere® Liberty configuration, see the related documentation, for example [HTTP Endpoint \(httpEndpoint\)](#) and [Configuring an httpEndpoint to use an SSL configuration](#).

Application server - changing the trace properties

To modify the trace properties, modify the `trace.xml` template as necessary.

Templates for the master domain manager are stored in the following paths:

On UNIX operating systems

```
TWA_home/usr/servers/engineServer/configDropins/templates
```

On Windows operating systems

```
TWA_home\usr\servers\engineServer\configDropins\templates
```

Templates for the Dynamic Workload Console are stored in the following paths:

On UNIX operating systems

```
DWC_home/usr/servers/dwcServer/configDropins/templates
```

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:

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`

To modify the `trace.xml` template, follow this procedure:

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.

Chapter 9. Administering an IBM i dynamic environment

On overview on how to administer the IBM Workload Scheduler IBM i dynamic environment.

To begin scheduling jobs with advanced options on IBM i agents, the agents must be configured.

Configuring the agent on IBM i systems

An overview on how to configure the agent on IBM i systems.

The configuration settings of the agent are contained in the `JobManager.ini` file and in the `JobManagerGW.ini` file (for the path of these files, see the section about installation paths in *IBM Workload Scheduler: Planning and Installation*).

The configuration files are made up of many different sections. Each section name is enclosed between square brackets and each section includes a sequence of `variable = value` statements.

You can customize properties for the following:

- Log properties
- Trace properties when the agent is stopped. You can also customize traces when the agent is running using the procedure described in [Configuring trace properties when the agent is running on page 84](#).
- Native job executor
- Java™ job executor
- Resource advisor agent
- System scanner

On IBM i systems, the log messages are written in the following file:

```
TWA_DATA_DIR>/stdlist/JM/JobManager_message.log
```

On IBM i systems, the trace messages are written in the following files:

```
<TWA_DATA_DIR>/TWS/stdlist/JM/ITA_trace.log  
<TWA_DATA_DIR>/TWS/stdlist/JM/JobManager_trace.log  
<TWA_DATA_DIR>/TWS/stdlist/JM/javaExecutor0.log
```

Not all the properties in the `JobManager.ini` file and in the `JobManagerGW.ini` file can be customized. For a list of the configurable properties, see the following sections:

- [Configuring log message properties \[JobManager.Logging.ccllog\] on page 81](#).
- [Configuring trace properties when the agent is stopped \[JobManager.Logging.ccllog\] on page 82](#).
- [Configuring common launchers properties \[Launchers\] on page 87](#).
- [Configuring properties of the native job launcher \[NativeJobLauncher\] on page 89](#).
- [Configuring properties of the Java job launcher \[JavaJobLauncher\] on page 92](#).
- [Configuring properties of the Resource advisor agent \[ResourceAdvisorAgent\] on page 92](#).
- [Configuring properties of the System scanner \[SystemScanner\] on page 95](#)



Note: In the `JobManager.ini` file and in the `JobManagerGW.ini` file you must refer to Java 64 bit version.

Configuring log message properties [JobManager.Logging.cclog]

About this task

To configure the logs, edit the [JobManager.Logging.cclog] section in the `JobManager.ini` file. This procedure requires that you stop and restart the IBM Workload Scheduler agent

The section containing the log properties is named:

```
[JobManager.Logging.cclog]
```

You can change the following properties:

JobManager.loggerhd.fileName

The name of the file where messages are to be logged. the default value is

On Windows operating systems

`POSIXHOME\stdlist\JM\JOBMANAGER-FFDC` where `POSIXHOME` is the installation directory.

On UNIX operating systems

`$(TWA_DATA_DIR)/stdlist/JM/JobManager_message.log`

JobManager.loggerhd.maxFileBytes

The maximum size that the log file can reach. The default is **1024000** bytes.

JobManager.loggerhd.maxFiles

The maximum number of log files that can be stored. The default is **3**.

JobManager.loggerhd.fileEncoding

By default, log files for the agent are coded in UTF-8 format. If you want to produce the log in a different format, add this property and specify the required codepage.

JobManager.loggerfl.level

The amount of information to be provided in the logs. The value ranges from 3000 to 7000. Smaller numbers correspond to more detailed logs. The default is **3000**.

JobManager.ffdc.maxDiskSpace

Exceeding this maximum disk space, log files collected by the first failure data capture mechanism are removed, beginning with the oldest files first.

JobManager.ffdc.baseDir

The directory to which log and trace files collected by the ffdc tool are copied. The default directory is

On Windows operating systems

`POSIXHOME\stdlist\JM\JobManager_message.log` where `POSIXHOME` is the installation directory.

On UNIX operating systems

`$(TWA_DATA_DIR)/stdlist/JM/JobManager_message.log`

JobManager.ffdc.filesToCopy

Log and trace files (`JobManager_message.log` and `JobManager_trace.log`) collected by the `ffdc` tool located in `<TWA_home>\TWS\stdlist\JM`. For example, `JobManager.ffdc.filesToCopy = "/opt/IBM/TWA_<TWS_user>/TWS/stdlist/JM/JobManager_message.log" "/opt/IBM/TWA_<TWS_user>/TWS/stdlist/JM/JobManager_trace.log"`

When a message is logged (`JobManager.ffdc.triggerFilter = JobManager.msgIdFilter`) that has an ID that matches the pattern "AWSITA*E" (`JobManager.msgIdFilter.msgIds = AWSITA*E`), which corresponds to all error messages, then the log and trace files (`JobManager.ffdc.filesToCopy = "/opt/IBM/TWA_<TWS_user>/TWS/stdlist/JM/JobManager_message.log" "/opt/IBM/TWA_<TWS_user>/TWS/stdlist/JM/JobManager_trace.log"`) are copied (`JobManager.ffdc.className = ccg_ffdc_filecopy_handler`) to the directory `JOBMANAGER-FFDC` (`JobManager.ffdc.baseDir = /opt/IBM/TWA_<TWS_user>/TWS/stdlist/JM/JOBMANAGER-FFDC`). If the files copied exceed 10 MB (`JobManager.ffdc.maxDiskSpace = 10000000`), then the oldest files are removed first (`JobManager.ffdc.quotaPolicy = QUOTA_AUTODELETE`).

After installing the z-centric agent or dynamic agent on Windows 2012, the `JobManager_message.log` might not be created. In this case, perform the following procedure:

1. Stop the agent.
2. Create a backup copy of `JobManager.ini`, and edit the original file by changing the row:

```
JobManager.loggerhd.className = ccg_multiproc_filehandler
```

to

```
JobManager.loggerhd.className = ccg_filehandler
```

3. Restart the agent.

Configuring trace properties when the agent is stopped [JobManager.Logging.cclog]

How to configure the trace properties when the agent is stopped.

To configure the trace properties when the agent is stopped, edit the `[JobManager.Logging]` section in the `JobManager.ini` file and then restart the IBM Workload Scheduler agent.

The section containing the trace properties is named:

```
[JobManager.Logging.cclog]
```

You can change the following properties:

JobManager.trhd.fileName

The name of the trace file.

JobManager.trhd.maxFileBytes

The maximum size that the trace file can reach. The default is 1024000 bytes.

JobManager.trhd.maxFiles

The maximum number of trace files that can be stored. The default is 3.

JobManager.trfl.level

Determines the type of trace messages that are logged. Change this value to trace more or fewer events, as appropriate, or on request from IBM Software Support. Valid values are:

DEBUG_MAX

Maximum tracing. Every trace message in the code is written to the trace logs.

INFO

All *informational*, *warning*, *error* and *critical* trace messages are written to the trace. The default value.

WARNING

All *warning*, *error* and *critical* trace messages are written to the trace.

ERROR

All *error* and *critical* trace messages are written to the trace.

CRITICAL

Only messages which cause the agent to stop are written to the trace.

The output trace (`JobManager_trace.log`) is provided in XML format.

After installing the z-centric agent or dynamic agent on Windows 2012, the `JobManager_trace.log` might not be created. In this case, perform the following procedure:

1. Stop the agent.
2. Create a backup copy of `JobManager.ini`, and edit the original file by changing the row:

```
JobManager.trhd.className = ccg_multiproc_filehandler
```

to

```
JobManager.trhd.className = ccg_filehandler
```

3. Restart the agent.

Configuring trace properties when the agent is running

Use the **twstrace** command to set the trace on the agent when it is running.

Using the **twstrace** command, you can perform the following actions on the agent when it is running:

- [See command usage and verify version on page 84.](#)
- [Enable or disable trace on page 84.](#)
- Set the traces to a specific level, specify the number of trace files you want to create, and the maximum size of each trace file. See [Set trace information on page 85.](#)
- [Show trace information on page 85.](#)
- Collect trace files, message files, and configuration files in a compressed file using the command line. See [Collect trace information on page 86.](#)
- Collect trace files, message files, and configuration files in a compressed file using the Dynamic Workload Console. See [Retrieving IBM Workload Scheduler agent traces from the Dynamic Workload Console.](#)

You can also configure the traces when the agent is not running by editing the [JobManager.Logging] section in the `JobManager.ini` file as described in [Configuring the agent](#) section. This procedure requires that you stop and restart the agent.

twstrace command

Use the **twstrace** command to configure traces, and collect logs, traces, and configuration files (`ita.ini` and `jobManager.ini`) for agents. You collect all the information in a compressed file when it is running without stopping and restarting it.

See command usage and verify version

To see the command usage and options, use the following syntax.

Syntax

```
twstrace -u | -v
```

Parameters

-u

Shows the command usage.

-v

Shows the command version.

Enable or disable trace

To set the trace to the maximum or minimum level, use the following syntax.

Syntax

```
twstrace -enable | -disable
```

Parameters

-enable

Sets the trace to the maximum level. The maximum level is **1000**.

-disable

Sets the trace to the minimum level. The minimum level is **3000**.

Set trace information

To set the trace to a specific level, specify the number of trace files you want to create, and the maximum size the trace files can reach, use the following syntax.

Syntax

```
twstrace [ -level <level_number> ] [ -maxFiles <files_number> ] [ -maxFileBytes <bytes_number> ]
```

Parameters**-level <level_number>**

Sets the trace level. Specify a value in the range from 1000 to 3000, which is also the default value. Note that if you set this parameter to 3000, you have the lowest verbosity level and the fewest trace messages. To have a better trace level, with the most verbose trace messages and the maximum trace level, set it to **1000**.

-maxFiles <files_number>

Specify the number of trace files you want to create.

-maxFileBytes <bytes_number>

Set the maximum size in bytes that the trace files can reach. The default is **1024000** bytes.

Show trace information

To display the current trace level, the number of trace files, and the maximum size the trace files can reach, use the following syntax.

Syntax

```
twstrace -level | -maxFiles | -maxFileBytes
```

Parameters**-level**

See the trace level you set.

-maxFiles

See the number of trace files you create.

-maxFileBytes

See the maximum size you set for each trace file

Example**Sample**

The example shows the information you receive when you run the following command:

```
twstrace -level -maxFiles -maxFileBytes
```

```
AWSITA176I The trace properties are: level="1000",
max files="3", file size="1024000".
```

Collect trace information

To collect the trace files, the message files, and the configuration files in a compressed file, use the following syntax.

Syntax

```
twstrace -getLogs [ -zipFile <compressed_file_name> ] [ -host <host_name> ] [ -protocol {http | https} ] [ -port <port_number> ] [ -iniFile <ini_file_name> ]
```

Parameters

-zipFile <compressed_file_name>

Specify the name of the compressed file that contains all the information, that is logs, traces, and configuration files (ita.ini and jobManager.ini) for the agent. The default is **logs.zip**.

-host <host_name>

Specify the host name or the IP address of the agent for which you want to collect the trace. The default is **localhost**.

-protocol http|https

Specify the protocol of the agent for which you are collecting the trace. The default is the protocol specified in the **.ini** file of the agent.

-port <port_number>

Specify the port of the agent. The default is the port number of the agent where you are running the command line.

-iniFile <ini_file_name>

Specify the name of the **.ini** file that contains the SSL configuration of the agent for which you want to collect the traces. If you are collecting the traces for a remote agent for which you customized the security certificates, you must import the certificate on the local agent and specify the name of the **.ini** file that contains this configuration. To do this, perform the following actions:

1. Extract the certificate from the keystore of the remote agent.
2. Import the certificate in a local agent keystore. You can create an ad hoc keystore whose name must be **TWSCClientKeyStore.kdb**.
3. Create an **.ini** file in which you specify:

- **0** in the **tcp_port** property as follows:

```
tcp_port=0
```

- The port of the remote agent in the **ssl_port** property as follows:

```
ssl_port=<ssl_port>
```

- The path to the keystore you created in Step 2 on page 446 in the **key_repository_path** property as follows:

```
key_repository_path=<local_agent_keystore_path>
```

Configuring common launchers properties [Launchers]

About this task

In the `JobManager.ini` file, the section containing the properties common to the different launchers (or executors) is named:

```
[Launchers]
```

The following properties are available:

BaseDir

The installation path of the IBM Workload Scheduler agent. Do not modify this value.

CommandHandlerMinThreads

Indicates the maximum number of commands that can be run on the agent concurrently. Limits to the number of jobs vary depending on the resources of your workstation, however consider that operations on comdhandler are usually short. The default is **20**. Usually, there is no need to modify this setting, even if you plan a very high workload on the agent. You might want to change it if many commands are run concurrently on the agent, for example, many concurrent requests to retrieve job logs.

CommandHandlerMaxThreads

Indicates the maximum number of commands that can be run on the agent concurrently. Limits to the number of jobs vary depending on the resources of your workstation, however consider that operations on comdhandler are usually short. The default is **100**. Usually, there is no need to modify this setting, even if you plan a very high workload on the agent. You might want to change it if many commands are run concurrently on the agent, for example, many concurrent requests to retrieve job logs.

CpaHeartBeatTimeSeconds

The polling interval in seconds used to verify if the **agent** process is still up and running. If the agent process is inactive the product stops also the **JobManager** process. The default is **30**. Modify only if you use dynamic pools with CPU-based requirements or optimization policies. With a lower value, the agent reacts quickly to CPU modifications, but this might cause unstable values in case of CPU spikes. Lower values causes a higher use of resources on the agent.

DirectoryPermissions

The access rights assigned to the agent for creating directories when running jobs. The default is **0755**. Supported values are UNIX-format entries in hexadecimal notation.

DownloadDir

The name of the directory where the fix pack installation package or upgrade elmage for fault-tolerant agents or dynamic agents is downloaded during the centralized agent update process. If not specified, the following default directory is used:

On Windows operating systems:

```
<TWA_home>\TWS\stdlist\JM\download
```

On UNIX operating systems:

```
<TWA_home>/TWS/stdlist/JM/download
```

The centralized agent update process does not apply to z-centric agents.

ExecutorsMaxThreads

Specifies the maximum number of jobs the dynamic agent can run concurrently. For example, to allow the dynamic agent to run a maximum of 500 jobs concurrently, set this parameter to **500**. The default is **400**.

ExecutorsMinThreads

Specifies the minimum number of jobs the dynamic agent can run concurrently. For example, to allow the dynamic agent to run a minimum of 500 jobs concurrently, set this parameter to **500**. The default is **38**. Modify if the number of expected concurrent jobs is much higher than 38. The agent dynamically allocates more threads if necessary, until it reaches the value specified in **ExecutorsMaxThreads**.

FilePermissions

The access rights assigned to the agent for creating files when running jobs. The default is **0755**. Supported values are UNIX-format entries in hexadecimal notation.

MaxAge

The number of days that job logs are kept (in path *TWA_home/TWS/stdlist/JM*) before being deleted. The default is **30**. Possible values range from a minimum of 1 day.

NotifierMaxThreads

Notifier threads are in charge of notifying the dynamic workload broker of each status change in a job. This parameter specifies the maximum number of job status changes that can be notified to the dynamic workload broker.

NotifierMinThreads

Notifier threads are in charge of notifying the dynamic workload broker of each status change in a job. This parameter specifies the minimum number of job status changes that can be notified to the dynamic workload broker. The default value is **3**. Modify this parameters only in case of unexpected errors and after consulting with software support team.

SpoolDir

The path to the folder containing the jobstore and outputs. The default is:

```
value of BaseDir/stdlist/JM
```

StackSizeBytes

The size of the operating system stack in bytes. The default is **DEFAULT**, meaning that the **agent** uses the default value for the operating system. Do not modify this parameter unless instructed to do so by the software support team. Incorrect values can cause the agent to crash.

Configuring properties of the native job launcher [NativeJobLauncher]

About this task

In the `JobManager.ini` file, the section containing the properties of the native job launcher is named:

```
[NativeJobLauncher]
```

You can change the following properties:

AllowRoot

Applies to UNIX™ systems only. Specifies if the root user can run jobs on the agent. It can be `true` or `false`. The default is `false`. This property does not apply to IBM i, use the `AllowQSECOFR` option instead.

AllowQSECOFR

Applies to IBM i systems only. Specifies if QSECOFR user can run jobs on the agent. It can be `true` or `false`. The default is `true`. Add a line like `AllowQSECOFR = false` to the `JobManager.ini` file to deny job execution to QSECOFR.

CheckExec

If `true`, before launching the job, the agent checks both the availability and the execution rights of the binary file. The default is `true`.

DefaultWorkingDir

Specifies the working directory of native jobs. You can also specify the value for the working directory when creating or editing the job definition in the Workload Designer. When specified in the Workload Designer, this value overrides the value specified for the `DefaultWorkingDir` property. If you do not specify any working directories, the `<TWS_home>\bin` directory is used.

JobUnspecifiedInteractive

Applies to Windows™ operating systems only. Specifies if native jobs are to be launched in interactive mode. It can be `true` or `false`. The default is `false`.

KeepCommandTraces

Set to `true` to store the traces of the method invocation for actions performed on a job definition, for example, when selecting from a picklist. These files are stored in the path `/opt/IBM/TWA_<TWS_user>/TWS/stdlist/JM/r3batch_cmd_exec`. The default setting is `false`.

KeepJobCommandTraces

Set to `true` to store the traces of the method invocation for actions performed on a job instance, for example, viewing a spool list. These files are stored in the `.zip` file of the job instance. The default setting is `true`.

LoadProfile

Applies to agents on Windows servers only. Specifies if the user profile is to be loaded. It can be `true` or `false`. The default is `true`.

MonitorQueueName

Specifies the name of the queue where the IBM i jobs are monitored. If you do not specify this property, the default queue (QBATCH) is used.

PortMax

The maximum range of the port numbers used by the task launcher to communicate with the Job Manager. The default is 0, meaning that the operating system assigns the port automatically.

PortMin

The minimum range of the port numbers used by the task launcher to communicate with the Job Manager. The default is 0, meaning that the operating system assigns the port automatically.

PostJobExecScriptPathName

The fully qualified path of the script file that you want to run when the job completes. By default, this property is not present in the `JobManager.ini` file. If you do not specify any file path or the script file doesn't exist, no action is taken.

This property applies to dynamic agent and z/OS agent. For details about running a script when a job completes, see *User's Guide and Reference*.

PromotedNice

Used in workload service assurance. This property is not supported on the Agent for z/OS.

For UNIX and Linux operating systems only, assigns the priority value to a critical job that needs to be promoted so that the operating system processes it before others. Applies to critical jobs or predecessors that need to be promoted so that they can start at their critical start time.

Boundary values vary depending upon each specific platform, but generally lower values correspond to higher priority levels and vice versa. The default is -1.

Be aware that:

- The promotion process is effective with negative values only. If you set a positive value, the system runs it with the -1 default value.
- An out of range value (for example -200), prompts the operating system to automatically promote the jobs with the lowest allowed nice value.
- Overusing the promotion mechanism (that is, defining an exceedingly high number of jobs as mission critical and setting the highest priority value here) might overload the operating system, negatively impacting the overall performance of the workstation.

PromotedPriority

Used in workload service assurance. This property is not supported on the Agent for z/OS.

For Windows operating systems only, sets to this value the priority by which the operating system processes a critical job when it is promoted. Applies to critical jobs or predecessors that need to be promoted so that they can start at their critical start time. Valid values are:

- High
- AboveNormal (the default)
- Normal
- BelowNormal
- Low OR Idle

Note that if you set a lower priority value than the one non-critical jobs might be assigned, no warning is given.

RequireUserName

When `true`, requires that you add the user name in the JSDL job definition.

When `false`, runs with the user name used by job manager, that is:

- `TWS_user` on UNIX™ and Linux™ systems
- The local system account on Windows™ systems

The default is `false`.

RunExecutablesAsIBMiJobs

If you set this property to `true`, you can define IBM i jobs as generic jobs without using the XML definition. Generic jobs are automatically converted to IBM i jobs. As a side effect, generic jobs cannot be run when this parameter is enabled (`RunExecutablesAsIBMiJobs=true`). There is no default value because this property is not listed in the `JobManager.ini` file after the agent installation.

If you set this property to `true`, ensure that the user you used to install the agent has been granted the `*ALLOBJ` special authority.

ScriptSuffix

The suffix to be used when creating the script files. It is:

`.cmd`

For Windows™

`.sh`

For UNIX™

VerboseTracing

Enables verbose tracing. It is set to `true` by default.

Configuring properties of the Java™ job launcher [JavaJobLauncher]

About this task

In the `JobManager.ini` file, the section containing the properties of the Java™ job launcher is named:

```
[JavaJobLauncher]
```

You can change the following properties:

JVMDir

The path to the virtual machine used to start job types with advanced options. You can change the path to another compatible Java™ virtual machine.

JVMOptions

The options to provide to the Java™ Virtual Machine used to start job types with advanced options. Supported keywords for establishing a secure connection are:

- `https.proxyHost`
- `https.proxyPort`

Supported keywords for establishing a non-secure connection are:

- `Dhttp.proxyHost`
- `Dhttp.proxyPort`

For example, to set job types with advanced options, based on the default JVM http protocol handler, to the unauthenticated proxy server called with name `myproxyserver.mycompany.com`, define the following option:

```
JVMOptions = -Dhttp.proxyHost=myproxyserver.mycompany.com -Dhttp.proxyPort=80
```

Configuring properties of the Resource advisor agent [ResourceAdvisorAgent]

About this task

In the `JobManager.ini` and `JobManagerGW.ini` files, the section containing the properties of the Resource advisor agent is named:

```
[ResourceAdvisorAgent]
```

You can change the following properties:

BackupResourceAdvisorUrls

The list of URLs returned by the IBM Workload Scheduler master in a distributed environment or by the dynamic domain manager either in a z/OS or in a distributed environment. The agent uses this list to connect to the master or dynamic domain manager.

CPUScannerPeriodSeconds

The time interval that the Resource advisor agent collects resource information about the local CPU. The default value is every 10 seconds.

FullyQualifiedHostname

The fully qualified host name of the agent. It is configured automatically at installation time and is used to connect with the master in a distributed environment or with the dynamic domain manager in a z/OS or in a distributed environment. Edit only if the host name is changed after installation.

NotifyToResourceAdvisorPeriodSeconds

The time interval that the Resource advisor agent forwards the collected resource information to the Resource advisor. The default value is every 119 seconds.

ResourceAdvisorUrl**JobManager.ini**

The URL of the master in a distributed environment, or of the dynamic domain manager in a z/OS or in a distributed environment, that is hosting the agent. This URL is used until the server replies with the list of its URLs. The value is `https://$(tdwb_server):$(tdwb_port)/JobManagerRESTWeb/JobScheduler/resource`, where:

\$(tdwb_server)

is the fully qualified host name of the master in a distributed environment or of the dynamic domain manager either in a z/OS or in a distributed environment.

\$(tdwb_port)

is the port number of the master in a distributed environment or of the dynamic domain manager either in a z/OS or in a distributed environment.

It is configured automatically at installation time. Edit only if the host name or the port number are changed after installation, or if you do not use secure connection (set to `http`). If you set the port number to zero, the resource advisor agent does not start. The port is set to zero if at installation time you specify that you will not be using the master in a distributed environment or the dynamic domain manager either in a z/OS or in a distributed environment.

In a distributed environment, if **-gateway** is set to either `local` or `remote`, then this is the URL of the dynamic agent workstation where the gateway resides and through which the dynamic agents communicate. The value is `https://$(tdwb_server):$(tdwb_port)/ita/JobManagerGW/JobManagerRESTWeb/JobScheduler/resource`, where:

\$(tdwb_server)

The fully qualified host name of the dynamic agent workstation where the gateway resides and through which the dynamic agent communicates with the dynamic workload broker.

\$(tdwb_port)

The port number of the dynamic agent workstation where the gateway resides and through which the dynamic agent communicates with the dynamic workload broker.

JobManagerGW.ini

In a distributed environment, if **-gateway** is set to `local`, then **ResourceAdvisorUrl** is the URL of the master or dynamic domain manager. The value is `https://$(tdwb_server):$(tdwb_port)/JobManagerRESTWeb/JobScheduler/resource`, where:

\$(tdwb_server)

The fully qualified host name of the master or dynamic domain manager.

\$(tdwb_port)

The port number of the master or dynamic domain manager.

ScannerPeriodSeconds

The time interval that the Resource advisor agent collects information about all the resources in the local system other than CPU resources. The default value is every 120 seconds.

The resource advisor agent, intermittently scans the resources of the machine (computer system, operating system, file systems and networks) and periodically sends an update of their status to the master or dynamic domain manager either in a z/OS or in a distributed environment.

The CPU is scanned every `CPUScannerPeriodSeconds` seconds, while all the other resources are scanned every `ScannerPeriodSeconds` seconds. As soon as one of the scans shows a significant change in the status of a resource, the resources are synchronized with the master in a distributed environment or the dynamic domain manager either in a z/OS or in a distributed environment. The following is the policy followed by the agent to tell if a resource attribute has significantly changed:

- A resource is added or deleted
- A string attribute changes its value
- A CPU value changes by more than `DeltaForCPU`
- A file system value changes by more than `DeltaForDiskMB` megabytes
- A Memory value changes by more than `DeltaForMemoryMB` megabytes

If there are no significant changes, the resources are synchronized with the IBM Workload Scheduler master in a distributed environment or with the dynamic domain manager either in a z/OS or in a distributed environment every `NotifyToResourceAdvisorPeriodSeconds` seconds.

Configuring properties of the System scanner [SystemScanner]

About this task

In the `JobManager.ini` file, the section containing the properties of the System scanner is named:

```
[SystemScanner]
```

You can change the following properties:

CPUSamples

The number of samples used to calculate the average CPU usage. The default value is 3.

DeltaForCPU

The change in CPU usage considered to be significant when it becomes higher than this percentage (for example, DeltaForCPU is 20 if the CPU usage changes from 10 percent to 30 percent). The default value is 20 percent.

DeltaForDiskMB

The change in use of all file system resources that is considered significant when it becomes higher than this value. The default value is 100 MB.

DeltaForMemoryMB

The change in use of all system memory that is considered significant when it becomes higher than this value. The default value is 100 MB.

Configuring to schedule job types with advanced options

About this task


You can define job types with advanced options by using the related configuration files. The options you define in the configuration files apply to all job types with advanced options of the same type. You can override these options when defining the job by using the Dynamic Workload Console or, if you are in a distributed environment, the **composer** command.

Configuration files are available on each dynamic agent in TWA_home/TWS/JavaExt/cfg for the following job types with advanced options:

Table 84. Configuration files for job types with advanced options

Job type	File name	Keyword
<ul style="list-style-type: none"> • Database job type • MSSQL Job 	DatabaseJobExecutor.properties	<p>Use the <code>jdbcDriversPath</code> keyword to specify the path to the JDBC drivers. Define the keyword so that it points to the JDBC jar files directory, for example:</p> <pre>jdbcDriversPath=c:\\mydir\\jars\\jdbc</pre> <p>The JDBC jar files must be located in the specified directory or its subdirectories. Ensure you have list permissions on the directory and its sub subdirectories.</p>

Table 84. Configuration files for job types with advanced options (continued)

Job type	File name	Keyword
		 Note: For the MSSQL database, use version 4 of the JDBC drivers.
Java™ job type	JavaJobExecutor.properties	Use the <code>jarPath</code> keyword to specify the path to the directory where the jar files are stored. This includes all jar files stored in the specified directory and all sub directories.
J2EE job type	J2EEJobExecutorConfig.properties	For more information about the J2EE job type, see the topic about configuring to schedule J2EE jobs in the <i>IBM Workload Scheduler: Administration Guide</i> .

Customizing the SSL connection between IBM i agents and a master domain manager or a dynamic domain manager using your own certificates

Customizing the SSL connection between a master domain manager or a dynamic domain manager and IBM i agents connected to it using your own certificates.

About this task

By default the communication between IBM i agents and a master domain manager or a dynamic domain manager to which they are registered uses the https protocol.

The SSL communication uses the default certificates provided by IBM Workload Scheduler.

The master domain manager uses two keystores in .jks format: a private key keystore and a trusted key keystore:

On Windows systems

Private keys keystore

```
TWA_home>\usr\servers\engineServer\resources\security
\TWSServerKeyFile.jks
```

Trusted keys keystore

```
TWA_home>\usr\servers\engineServer\resources\security
\TWSServerTrustFile.jks
```


On UNIX systems

Private keys keystore

```
TWA_DATA_DIR>/usr/servers/engineServer/resources/security/  
TWSServerKeyFile.jks
```

Trusted keys keystore

```
TWA_DATA_DIR>/usr/servers/engineServer/resources/security/  
TWSServerTrustFile.jks
```

If you want to use your own customized certificates for this communication because you customized the master domain manager or the dynamic domain manager certificates, you must customize the agent certificates and the agent configuration file.

To enable communication between a master domain manager or a dynamic domain manager and an IBM i agent, you must first create your own certificates for IBM i agent and then trust the agents certificates in the master domain manager or the dynamic domain manager keystore.

Perform the following steps:

1. Log on as Administrator on Windows operating systems or as root on UNIX and Linux operating systems, on the machine where you installed a IBM Workload Scheduler instance that contains the openssl utility, for example, the master domain manager or the dynamic domain manager.
2. Go to the `TWS_INST_DIR/TWS/ssl` directory, where `TWS_INST_DIR` is the IBM Workload Scheduler installation directory and copy there the following files:
 - `TWS_INST_DIR/TWS/bin/openssl(.exe)`
 - `TWS_INST_DIR/TWS/bin/openssl.cnf`
3. Generate a random file for the IBM i agent, by using the following command:

```
openssl rand  
-out suffix.rnd  
-rand ./openssl 8192
```

where `suffix` is a generic word. For example, you can use the IBM i agent workstation name to easily find the files generated for this workstation.

4. Generate the `suffix.key` private key, by running the following command:

```
openssl genrsa -des3  
-out suffix.key 2048
```

and save the password that you entered in the previous command in the `suffix.pwd` file.



Note: Ensure that you take note of the password you insert because you need it in the following steps.

5. Generate the `ita_prvsuffix.pem` PEM file containing the agent private key, by renaming the `suffix.key` in `ita_prvsuffix.pem`.
6. Save the agent private key password in a `suffix.sth` stash file by using the following command:

```
openssl base64
-in suffix.pwd
-out suffix.sth
```

7. Generate the `suffix.csr` certificate signature request by running the following command:

```
openssl req -new
-key suffix.key
-out suffix.csr
-config ./openssl.cnf
```

8. Generate the `suffix.crt` certificate that contains the private key `suffix.key` by running the following command:

```
openssl x509 -req
-CA TWScA.crt
-CAkey TWScA.key
-days 365
-in suffix.csr
-out suffix.crt
-Ccreateserial
```

9. Generate the `suffix.pem` PEM file containing the agent private key certificate by creating a copy of the `suffix.crt` certificate, and name the copied file `suffix.pem`.
10. Generate the `ita_pubsuffix.pem` PEM file containing the agent private key certificate by creating a copy of the `suffix.crt` certificate, and name the copied file `ita_pubsuffix.pem`.
11. Create a copy of the `ita_pubsuffix.pem` file created in step 10 on page 458 and name the copied file `ita_certsuffix.pem`.
12. On the master domain manager or the dynamic domain manager machine to which the IBM i agent is to be connected, generate the `server.pem` certificate by running the command:

```
keytool -export -rfc
-alias server
-file TWS_INST_DIR/TWS/ssl/server.pem
-keypass password
-keystore path/TWSServerKeyFile.jks
-storepass default
```

where `password` is the value you entered in step 4 on page 457 and `path` is the path listed at the beginning of this topic.

13. Generate the `ita_ca_certsuffix.pem` file which is the concatenation of the `ita_pubsuffix.pem` and of the `server.pem` files, by performing the following actions:
- Create a copy of the `ita_pubsuffix.pem` file and name it `ita_ca_certsuffix.pem`.
 - Edit the `ita_ca_certsuffix.pem` file.
 - Append at the end of the `ita_ca_certsuffix.pem` file content the `server.pem` file content.
 - Save the final version of the `ita_ca_certsuffix.pem` file.



Note: The `ita_ca_certsuffix.pem` file contains the certificates of the IBM i agent and the master domain manager or the dynamic domain manager to which the agent is connected.

14. Log on as `TWS_IBMi_USER` user on the IBM i agent machine and locate the `TWS_IBMI_INSTDIR/TWS/ITA/cpa/ita/cert/` directory where `TWS_IBMI_INSTDIR` is the directory where you installed the IBM Workload Scheduler IBM i agent for the `TWS_IBMi_USER` user.
15. From the `TWS_INST_DIR/TWS/ssl` directory of the machine where you generated the PEM files, copy into the `TWS_IBMI_INSTDIR/TWS/ITA/cpa/ita/cert/` directory of the IBM i agent installation directory the following files:
 - `ita_prvsuffix.pem`.
 - `ita_pubsuffix.pem`.
 - `ita_certsuffix.pem`.
 - `ita_ca_certsuffix.pem`.
 - `suffix.sth`.
 - `suffix.rnd`.



Note: Ensure that the files you copied have `TWS_IBMi_USER` ownership.

16. On the machine where you installed the IBM i agent, open the `ita.ini` configuration agent file and set the values appropriate for your environment in the following properties: Where:

`stash_file_fullpath`

Specify the fully qualified path to the `suffix.sth` stash file that contains the agent private key password. This is the file you created in step 6 on page 457. The default value is `TWS_IBMI_INSTDIR/TWS/ITA/cpa/ita/cert/password.sth`.

`random_file_fullpath`

Specify the fully qualified path to the `suffix.rnd` random file. This is the file that you created in step 3 on page 457. The default is `TWS_IBMI_INSTDIR/TWS/ITA/cpa/ita/cert/TWS.rnd`.

`label_agent_private_key`

Specify the label of the agent private key.

`suffix`

Specify the suffix that you used in the names of all the files that you generated. The default value is `tws`.

`directory_ita_*suffix.pem>`

Specify the directory that contains the following `.pem` files that you generated:

Truststore

`ita_ca_certsuffix.pem` that you generate in step 13 on page 458

Keystore

- `ita_prvsuffix.pem` that you generated in [step 5 on page 457](#).
- `ita_pubsuffix.pem` that you generated in [step 10 on page 458](#).
- `ita_certsuffix.pem` that you generated in [step 11 on page 458](#).

The default directory is `TWS_IBMI_INSTDIR/TWS/ITA/cpa/ita/cert`.

17. Stop the IBM i agent by using the following command:

```
ShutDownLwa
```

18. Start the IBM i agent by using the following command:

```
StartUpLwa
```

19. On the master domain manager or the dynamic domain manager machine which the IBM i agent is to be connected to, trust the `TWS_INST_DIR/TWS/ssl/suffix.pem` IBM i agent certificate that you generated in [step 9 on page 458](#), in the keystore, by running the following steps:

```
keytool -import -trustcacerts
-alias <suffix>
-file <TWS_INST_DIR>/TWS/ssl/<suffix>.pem
-keypass <password>
-keystore <path>/
    TWSServerTrustFile.jks
-storepass default
```

where `TWS_INST_DIR` is the master domain manager or the dynamic domain manager installation directory and `<password>` is the value you entered in [step 4 on page 457](#). `<path>` is the path listed at the beginning of this topic.

Example

You have the following environment:

- IBM i agent installed in the `opt/ibm/TWS` directory of the `nc117031` machine for the user `twuserIBMi`.
- Master domain manager installed in the `opt/IBM/TWA92` directory of the machine `nc060201`.

To create the IBM i agent certificates to connect to the master domain manager, perform the following steps:

1. Log on as root on the `nc060201` machine where you installed the master domain manager.
2. Go to the `opt/IBM/TWA92/TWS/ssl` directory and copy there the following files:
 - `opt/IBM/TWA92/TWS/bin/openssl`
 - `opt/IBM/TWA92/TWS/bin/openssl.cnf`
3. Generate the `nc117031.rnd` random file in the `opt/IBM/TWA92/TWS/ssl` directory by running the following command:

```
openssl rand
-out nc117031.rnd
-rand ./openssl 8192
```

4. Generate the `nc117031.key` private key in the `opt/IBM/TWA92/TWS/ssl` directory by running the following command:

```
openssl genrsa -des3
-out nc117031.key 2048
```

and save the `maestro00` password that you entered in the `nc117031.pwd` file in text format in the `opt/IBM/TWA92/TWS/ssl` directory.

5. Create a copy of the `nc117031.key` file in the `opt/IBM/TWA92/TWS/ssl` directory and name it `ita_prvnc117031.pem`.
6. Save the `maestro00` password in a `nc117031.sth` stash file in the `opt/IBM/TWA92/TWS/ssl` directory by running the following command:

```
openssl base64
-in nc117031.pwd
-out nc117031.sth
```

7. Generate the `nc117031.csr` certificate signature request in the `opt/IBM/TWA92/TWS/ssl` directory by running the following command:

```
openssl req -new
-key nc117031.key
-out nc117031.csr
-config ./openssl.cnf
```

8. Generate the `nc117031.crt` certificate in the `opt/IBM/TWA92/TWS/ssl` directory that contains the private key `nc117031.key` by running the following command:

```
openssl x509 -req
-CA TWSca.crt
-CAkey TWSca.key
-days 365
-in nc117031.csr
-out nc117031.crt
-CACreateserial
```

9. Create a copy of the `nc117031.crt` certificate in the `opt/IBM/TWA92/TWS/ssl` directory and name it `nc117031.pem`.
10. Create a copy of the `nc117031.crt` certificate in the `opt/IBM/TWA92/TWS/ssl` directory and name it `ita_pubnc117031.pem`.
11. Create a copy of the `ita_pubnc117031.pem` file in the `opt/IBM/TWA92/TWS/ssl` directory and name it `ita_certnc117031.pem`.
12. On the `nc060201` machine, generate the `server.pem` certificate in the `opt/IBM/TWA92/TWS/ssl` directory by running the following command:

```
keytool -export -rfc
-alias server
-file opt/IBM/TWA/TWS/ssl/server.pem
-keypass maestro00
-keystore path>/TWSServerKeyFile.jks
-storepass default
```

where `path` is the path listed at the beginning of this topic.

13. Generate the `ita_ca_certnc117031.pem` file in the `opt/IBM/TWA/TWS/ssl` directory which is the concatenation of the `ita_pubnc117031.pem` and the `server.pem` files, by performing the following actions:

- a. Create a copy of `ita_pubnc117031.pem` file in the `opt/IBM/TWA/TWS/ssl` directory and name it `ita_ca_certnc117031.pem`.
 - b. Edit the `ita_ca_certnc117031.pem` file.
 - c. Append at the end of the `ita_ca_certnc117031.pem` file content the `server.pem` file content.
 - d. Save the final version of the `ita_ca_certnc117031.pem` file.
14. Log on as `twuserIBMi` user on the `nc117031` machine and locate the `opt/ibm/TWS/ITA/cpa/ita/cert/` directory.
 15. From the `opt/IBM/TWA/TWS/ssl` directory of the `nc060201` machine where you generated the PEM files, copy into the `opt/ibm/TWS/ITA/cpa/ita/cert/` directory the following files:
 - `ita_prvnc117031.pem`.
 - `ita_pubnc117031.pem`.
 - `ita_certnc117031.pem`.
 - `ita_ca_certnc117031.pem`.
 - `nc117031.sth`.
 - `nc117031.rnd`.

Ensure that all the files have `twuserIBMi` ownership.

16. On the `nc117031` machine, open the `ita.ini` configuration agent file and set the following values for the listed properties:

```
password_file=opt/ibm/TWS/ITA/cpa/ita/cert/nc117031.sth
random_file=opt/ibm/TWS/ITA/cpa/ita/cert/nc117031.rnd
cert_label=nc117031
key_db_name=nc117031
key_repository_dir=opt/ibm/TWS/ITA/cpa/ita/cert/*nc117031.pem
```

17. Stop the IBM i agent by using the following command:

```
ShutDownLwa
```

18. Start the IBM i agent by using the following command:

```
StartupLwa
```

19. On the `nc060201` machine, trust the `opt/IBM/TWA92/TWS/ssl/nc117031.pem` agent certificate by running the following steps:

```
keytool -import -trustcacerts
-alias nc117031
-file opt/IBM/TWA/TWS/ssl/ssl/nc117031.pem
-keypass maestro00
-keystore path>/TWSServerTrustFile.jks
-storepass default
```

where *path* is the path listed at the beginning of this topic.

Chapter 10. Performance

This chapter provides information about issues that impact performance. Use this information both to prevent problems occurring and to help resolve problems that occur.

Network traffic

A full description of how a IBM Workload Scheduler network is structured, and how the different nodes communicate, is provided at the beginning of [Network administration on page 264](#). In particular, see [Optimizing the network on page 278](#), which explains how to design and operate your IBM Workload Scheduler network to maximize performance.

Tracing

The performance of any workstation can be impacted by the level of tracing it has to perform. The *Troubleshooting Guide* has a chapter which explains the diagnostic tools that are available, and within that chapter there is a section about the IBM Workload Scheduler In-flight Tracing utility, which, as well as discussing how the feature works, also describes how to customize it to enhance workstation performance.

The performance might also be impacted by the tracing activities on WebSphere Application Server Liberty Base.

Logging

The performance of any workstation can be impacted by the way the IBM Workload Scheduler logging mechanism uses memory. The default settings applied in this version are designed to ensure the maximum performance. However, because these defaults are different from the defaults in earlier versions, if you are experiencing performance problems, it is advisable to check that these settings have not been in some way overwritten by the previous values. In the diagnostic tools chapter of *IBM Workload Scheduler: Troubleshooting Guide*, there is a section about CCLog, which, apart from discussing how to customize CCLog, also describes how to check the CCLog processing defaults.

Maintaining the database

Maintaining the database in a good state of organization is important to optimize performance. See [Reorganizing the database on page 342](#) for details.

Symphony file sizing

To calculate the size of the Symphony file and understand its impact on performance, see [Avoiding full file systems on page 343](#).

Tuning a UNIX™ domain manager to handle large numbers of fault-tolerant agents

The performance of domain managers on UNIX™ is impacted if they are overloaded with jobs. Improvements can be obtained by modifying the kernel parameters. The precise settings differ according to operating system, and you might need to test different settings to obtain optimum performance.

The following is an example of the kernel settings for Linux (kernel t3.10.0-514.e17.x86_64) to handle 500000 jobs per day workload:

```
data seg size=unlimited
scheduling priority=0
file size=unlimited
pending signals=124946
max locked memory=64
max memory size=unlimited
open files=105000
pipe size=8
POSIX message queues=819200
real-time priority=0
stack size=10240
cpu time=unlimited
max user processes=16384
virtual memory=unlimited
file locks=unlimited
```

Tuning job processing on a workstation

This section explains how to tune selected options in the IBM Workload Scheduler `localopts` file to improve IBM Workload Scheduler performance. These options control the period between successive instances of an activity. [Table 85: Options for tuning job processing on a workstation on page 464](#) shows the activities to be tuned, the corresponding option that can be set in the `localopts` file, and how the changed value impacts performance.

Table 85. Options for tuning job processing on a workstation

Activity	Option	Impact on performance
batchman periodically scans the <code>Symphony</code> file for jobs ready to be processed.	bm look	In all these cases, a shorter time means more frequent scans, using more cpu resources, and impacting other processes that are running. However, it also means that for all activities waiting time is kept to a minimum. If throughput is important and the workstation has plenty of memory, try shortening the times.
jobman accesses the <code>Courier.msg</code> file to see if there are jobs that need to be launched.	jm read	
After having launched a job jobman checks periodically for job completion status.	jm look	
mailman looks periodically in the <code>Mailbox.msg</code> for completed jobs.	mm read	A longer period between successive activities means jobs take longer to run, because there are longer waits for each activity. However, the reduced frequency of the scans means that more memory is available for jobs because less is being used by these monitoring activities.
batchman checks periodically in <code>Intercom.msg</code> for jobs that are complete so that it can update the <code>Symphony</code> file.	bm read	

Consider the meaning of the various options. If your objective is to run the jobs as quickly as possible, but you are not concerned about how quickly the

Table 85. Options for tuning job processing on a workstation (continued)

Activity	Option	Impact on performance
		information about completed jobs is distributed, you could reduce the wait periods for <i>bm look</i> and <i>jm read</i> , but increase the periods for the others.
		Alternatively, to speed up the overall job processing time (from initial job launch to the update with the completion status), you can tune <i>bm look</i> , <i>jm look</i> , and <i>mm read</i> .

If you decide to tune these setting do the following:

- Test the result in a test system before applying changes in your production environment. To get worthwhile results, the test environment must have the same characteristics as the production environment.
- Modify only the parameters that are necessary. It is better to modify one at a time and thoroughly test the change in performance, rather than changing all at once.
- Make a backup copy of the `localopts` file to ensure you can revert to the default options if necessary.

Stop and start the agent to activate changes applied to the `localopts` file.

Tuning plan replication

Tuning plan replication involves configuring specific settings to optimize the process of replicating plan data into the database. Plan replication ensures quick and reliable access to plan data stored in the database. Its main objective is to provide quick response times and increased overall performance. Sometimes, if this synchronization process is not configured appropriately for the size of your workload, you might notice some discrepancies in your environment, such as job status misalignment between the command line (conman) and the monitoring results obtained in the Dynamic Workload Console.

There are a few simple settings you can configure to optimize performance:

Configure the cache size

Add the following properties to the `TWSConfig.properties` file located in the following paths:

On Windows operating systems

```
<TWA_home>\usr\servers\engineServer\resources\properties
```

On UNIX operating systems

```
<TWA_DATA_DIR>/usr/servers/engineServer/resources/properties
```

```
#Custom property which defines the number of threads and queues needed to
handle the plan updates
com.ibm.tws.planner.monitor.subProcessors=10
```

```
#Custom property which optimizes the DB access for file
dependency status update
com.ibm.tws.planner.monitor.filecachesize=40000
```

```
#Customer property which optimizes the DB access for
job and job stream status update
com.ibm.tws.planner.monitor.cachesize=40000
```

In addition, follow the steps to increase the heap size settings (initialHeapSize = 2048 and maximumHeapSize = 4096) of the application server on the master domain manager as documented in the *Administration Guide*.

Tuning the database

To learn about tuning the database, consult the relevant product documentation:

DB2

Go to the DB2 Knowledge Center, and search for **Best practices**.

Oracle

See the *Performance Tuning Guide* in the Oracle documentation set.

Optimizing the replication of the Symphony file in the database

Tuning DB2 database configuration parameters to improve performance when the Symphony file is replicated in the database.

In a IBM Workload Scheduler environment where more than 200,000 jobs are scheduled to be submitted, there are several DB2 database configuration parameters that can be tuned to improve performance when the `Symphony` plan is replicated in the IBM Workload Scheduler database.

The following are the suggested values for a plan with more than 200,000 jobs:

```
LOGBUFSZ = 2150
DBHEAP = AUTOMATIC (or greater than LOGBUFSZ)

LOGFILSIZ = 3000
LOGPRIMARY = 200
LOGSECOND = 40

PAGE_AGE_TRGT_MCR = 120
```

In addition, increase the number of pages (NPAGES) of the `TWS_PLN_BUFFPOOL` parameter to 182000 and the `TWS_BUFFPOOL` parameter to 50000 by using the `ALTER BUFFERPOOL` command.

Before changing any of these values, refer to the information about tuning a DB2 database in the relevant product documentation in the DB2 Knowledge Center.

Tuning the WebSphere Application Server Liberty Base

To learn about tuning the WebSphere Application Server Liberty Base, consult the appropriate documentation.

Go to https://www.ibm.com/support/knowledgecenter/en/SSEQTP_liberty/com.ibm.websphere.wlp.doc/ae/twlp_tun.html.

Inadequate Java™ heap size

The default Java™ maximum heap size might be too small for your requirements. If you have any reason to suspect the performance of the WebSphere Application Server Liberty Base, increase the heap size as described in [Increasing application server heap size on page 472](#).

Too many manual job submissions

IBM Workload Scheduler is designed for maximum efficiency when handling jobs submitted using a scheduled plan. Consequently, it is less adapted to processing manually submitted jobs. Thus, performance can be improved by reducing the number of manually submitted jobs.

Too many file dependency checks

Each file dependency check has an impact on performance. If you design a plan that is constantly checking many file dependencies, you reduce the performance of the workstation where these jobs are being run.

If multiple “opens? files are being used as a dependency, use the “-a? (and) option. For example, to check if three home directories `/tom`, `/dick`, and `/harry` exist, before launching `myjob` issue the following:

```
job2 opens "/users" (-d %p/tom -a -d %p/dick -a -d %p/harry)
```

This checks for all three directories at the same time, instead of looking for each directory separately.

Other factors that impact performance when evaluating file dependencies are the `bm check` parameters in the `localopts` file. These are documented in the Localopts summary section in the Administration Guide.



Note: In case of file dependencies applied to dynamic agent, it is suggested to keep ratio *number of file dependencies/bm check file* less than 0.7

Network configuration availability

After a system reboot, network services might be slow to start and if the agent starts when network services are not yet ready, the agent cannot work properly.

To prevent this problem, the agent waits sixty seconds and then retries to retrieve the network configuration. The agent repeats this operation for 5 times, that is, it waits for a total of 5 minutes for the network configuration to become available. If all attempts fail, the agent stops working.

You can configure the number of times the agent waits for network configuration availability in the **ITA** section of the `ita.ini` file, as follows:

1. Browse to the path where the `ita.ini` file is located:

On UNIX™ operating systems

`TWA_DATA_DIR/ITA/cpa/ita/ita.ini`

On Windows™ operating systems

`TWA_homeTWS\ITA\cpa\config\ita.ini`

2. Edit the setting for the `net_conf_wait` parameter defining the number of times the agent retries to retrieve the network configuration, waiting sixty seconds between each attempt. If the number of attempts you have defined expires without the agent being able to retrieve the network configuration, the agent stops working.

Workload spreading

Whatever jobs you have to schedule, try and spread them out through the production period so that there is no concentration in any one moment. Try also to avoid scheduling activities during times when normal user traffic in the network is very heavy, for example during the morning when users commence working and deal with accumulated emails.

Failure to do this might cause a bottleneck at the Mailbox.msg queue, which causes delays in updating the Symphony file, which in turn creates delays in the availability of job statuses to conman, the Dynamic Workload Console.

Improving job-processing performance

The processing and monitoring of jobs on a workstation is controlled primarily by various parameters in the `localopts` file and the global options maintained by `optman`. These parameters are described in the *IBM Workload Scheduler: Planning and Installation Guide*.

If you are experiencing problems of performance when processing and monitoring jobs, contact IBM Software Support for advice about how to tune these parameters in your particular environment to improve performance.

Mailbox caching - advantages and disadvantages

Mailman uses a parameter in the `localopts` file to decide whether to cache mailbox messages: `mm cache mailbox`. This section explains the advantages and disadvantages of the on and off settings of this parameter.

Setting the `mm cache mailbox` parameter to `no`

This means that mailman has to make a separate read action for each message before processing it, and then a separate delete action after successfully processing the message. The I/O activity in performing these activities one message at a time is proportionally high for the amount of data being read. This has an impact on performance. On the other hand, the processing is simple, in that each message is read, processed, and then removed from the mailbox. Any failure of the system at any point means that at most one message is replayed and no data is lost.

Setting the `mm cache mailbox` parameter to `yes` (default)

This means that mailman reads a block of messages into cache memory, processes all of the messages, and then deletes all of them from the mailbox. The advantage in I/O time is clear; reading and deleting a sequential

set of messages in one action is a much more efficient use of I/O time, than reading and deleting them one-by-one, meaning improved performance.

However, if there is a failure of mailman or the operating system, the cache is lost. On restarting, mailman rereads the set of messages that were previously in cache, some of which might already have been processed. For example, if mailman reads a block of 32 messages into cache and has processed 30 of them when a problem occurs, when mailman is restarted it rereads those 32 records and has to process 30 duplicates before being able to continue where it stopped.

Most events deal with job state changes, and these events can be repeated without creating any problems, and the critical events mechanism is able to deal with the others. However, there is an impact on performance while this recovery processing is going on, and if the in-built mechanisms cannot handle the message duplication, a more serious error might occur, ultimately involving the full or partial loss of the mailbox contents.

The number of messages being read in one action is configurable, using the parameter *mm cache size*. The default value for this parameter is 32 messages, and the maximum is 512. Setting this parameter to a value higher than the default increases performance during correct working, but decreases the performance in the event of a failure, for the reasons stated above. In addition, the additional cache means that the memory required by the IBM Workload Scheduler engine also increases. If you have a workstation with limited memory, or memory-heavy applications running, it might be counterproductive to increase the mailbox cache because the operating system might have to start paging the cache memory.

In conclusion, the default setting maximizes performance; only if you start losing events should you set it to *no*.

Setting the synch level parameter

This section describes the impact of the different settings of the *synch level* parameter in the `localopts` file. The *synch level* parameter only impacts UNIX™ environments.

The I/O activity performed by the IBM Workload Scheduler engine in managing plans, job streams, and jobs, consists in reading from and writing to the `Symphony` file and the event files (`Mailbox.msg`, `Intercom.msg`, and `Courier.msg`). When IBM Workload Scheduler writes to these files it has more than a straightforward *write* operation to perform. For example, when it writes to the `Mailbox.msg` file it performs the actions described in the following pseudo code:

```
TWS_write_event_lock {
    Lock Mailbox to write
}

TWS_write_event_update {
    Check Available Space
    Write Header
    Write Record
    Update Write Pointer
    Unlock Mailbox
}
```

Each action requires one or more write accesses to the disk. The way these actions are performed with the different synch level options is as follows:

synch level = high

Each write operation on the event files is immediately physically written to disk. This has a heavy impact on performance caused by the high I/O dependency.

synch level = medium

Each write event is considered as a single operation. For example, while `TWS_write_event_lock` contains only one action, `TWS_write_event_update` comprises five actions. With `synch_level` at *medium*, the five actions in this write event would be completed in one physical disk access, thus drastically reducing the I/O overhead.

synch level = low (default)

The operating system decides how and when to synchronize the data to disk. The impact of this option is more difficult to assess, because the rules are different for each operating system and file system.

The fault-tolerant switch manager - impact on performance

This section describes the impact that the enablement of the fault-tolerant switch manager feature has on the performance of the general architecture and the individual system. The fault-tolerant switch manager is enabled by setting the `enSwfaultTo1` global option to *yes*. When it is set, the master domain manager distributes messages to all fault-tolerant agents with *FullStatus* set to *yes*. This option has not dynamic capabilities and is not designed to work with broker agents.

Enabling this option impacts the following:

- Network traffic
- Disk space



Note: The fault-tolerant switch manager facility is only available if all of the workstations in the domain are at version 8.2, fix pack level 4, or higher.

Network Traffic

Network traffic is unchanged under normal conditions, but is increased during the replay phase, according to your choice and only under special conditions.

The replay phase is an essential part of the processing performed by the `switchmgr` command. It occurs when the new domain manager processes its Symphony file against its copies of the messages received, as it attempts to update its copy of the Symphony file.

Under normal conditions, the outbound reliability does not create any additional network traffic, because the messages are only stored for an eventual replay operation. The multiple inbound connections do not generate additional traffic because the traffic that was previously copied by the domain manager to the *FullStatus* member is now copied to the *FullStatus* members directly by the fault-tolerant agents.

During the replay phase, the connection protocol initiated by mailman on the backup domain manager includes a new phase for the replay of messages not sent by the failed domain manager. The impact of the message replay might be important, depending on the number of messages "trapped" in the old domain manager.

Disk Space

There are two places within the network where disk space use increases following the activation of the additional fault tolerance.

These places are as follows:

- On the single fault-tolerant agent. Here, in addition to the `tomaster.msg` queue, new queues are created for the other *FullStatus* fault-tolerant agents. These queues need not be considered, because the impact on a single agent is small.
- On the *FullStatus* fault-tolerant agents acting as backup domain managers. Here new `ftbox` message files are created. Upward traffic to the upper domain manager is in `ftbox/ftup.msg` and downward traffic to the lower domain manager is in `ftbox/ftdown.msg`.

Scalability

In an environment with large numbers of scheduling objects, the following impacts are felt:

- [Impact on JnextPlan on page 471](#)
- [Impact on reporting on page 472](#)
- [Impact on event rule deployment on page 472](#)

The resolution for these problems often includes making the following changes:

- [Increasing application server heap size on page 472](#)
- [Increasing maximum DB2 log capacity on page 473](#)

Impact on JnextPlan

The main impact on performance caused by a large network of workstations running many jobs over a production period of many days, is on JnextPlan. The key factor is the number of job stream instances that JnextPlan needs to handle. JnextPlan has to process each of these instances, and the time it takes to do so is a factor that can only be reduced by ensuring that the master domain manager and the database are on the most powerful computers possible, and that the communication, whether in local or remote, between the master domain manager and the database is maximized.

However, there are some specific measures that need to be taken as the number of jobs or job stream instances increases:

Number of jobs in the plan exceeds 40 000

In this event you need to increase the Java™ heap size used by the application server. The default is 512 MB, and you should at least double the heap size when job numbers exceed this level. Follow the procedure in [Increasing application server heap size on page 472](#).

You have a large number of job stream instances in the plan

DB2®

The default DB2® transaction log files cannot handle more than the transactions generated by about 180 000 job stream instances. You need to change the parameters that control the log file sizes or the numbers of log files that can be created, or both. Follow the procedure in [Increasing maximum DB2 log capacity on page 473](#).

Oracle

The number of transactions that can be managed by the Oracle log files depends on the way the Oracle database is configured. See the Oracle documentation for more details.



Note: If circumstances change and the number of job stream instances handled by JnextPlan falls below about 180 000, consider resetting the log and application server heap size settings to their default values, to avoid performance problems.

Impact on reporting

When a report is being processed, extra memory is required to handle large numbers of scheduling objects. The critical point is approximately 70 000 objects. This problem can be handled by increasing the Java™ heap size used by the application server. Follow the procedure in [Increasing application server heap size on page 472](#).

Impact on event rule deployment

When deploying large numbers of event rules, extra memory is required. The critical point is approximately 8 000 rules. This problem can be handled by increasing the Java™ heap size used by the application server. Follow the procedure in [Increasing application server heap size on page 472](#).

Increasing application server heap size

Follow this procedure to increase the Java™ heap size:

1. Log on to the computer where IBM Workload Scheduler is installed as the following user:

On Windows™ operating systems:

Any user in the *Administrators* group.

On UNIX™ operating systems:

root

2. Stop the WebSphere Application Server Liberty Base either by using the conman stopappserver command (see [Starting and stopping the application server and appservman on page 436](#)) or by running:

On Windows™ operating systems:

```
<TWA_home>\server_wauser\appservertools\stopAppServer.bat
```


On UNIX™ operating systems:

```
<TWA_home>/server_wauser/appservertools/stopAppServer.sh
```

- Open the following file:

On Windows™ operating systems:

```
<TWA_home>\server_wauser\usr\servers\engineServer\configDropins\overrides
\jvm.options
```

On UNIX™ operating systems:

```
<TWA_DATA_DIR>/server_wauser/usr/servers/engineServer/configDropins/
overrides/jvm.options
```

- Edit it as follows:

```
-Xms4096m
-Xmx4096m
-Xgcpolicy:gencon
#nursery mem size
-Xmn1024m
```



Note: In case of high workload (more than 200000 jobs/day) use 6144 as heap size and 1536 as nursery mem size. The above suggested settings must be applied when the RAM configuration value twice the value of the heap size.

- Save the file `jvm.option`
- Start the WebSphere Application Server Liberty Base, either by using the `conman startappserver` command (see [Starting and stopping the application server and appservman on page 436](#)) or by running

Windows operating systems:

```
<TWA_home>\server_wauser\appservertools\stopAppServer.bat
```

UNIX operating systems:

```
<TWA_home>/server_wauser/appservertools/stopAppServer.sh
```

Increasing maximum DB2® log capacity

The IBM Workload Scheduler DB2® database uses a transaction log the maximum size of which is fundamentally important for the successful running of JnextPlan on very large databases.

The default log consists of 40 primary log files, which are always present, and 20 secondary log files, created on demand. Each file is about 4 MB in size, so the maximum log capacity using all of the "secondary" log files as well as the primary files is $(40 + 20) \times 4 \text{ MB} = 240 \text{ MB}$.

The log space used by JnextPlan is dependent on the size of the preproduction plan. Approximately every 1000 job stream instances generate transactions that occupy 1 MB of space in the log file. Thus, the log files by default have a maximum

theoretical capacity of 240 000 job stream instances. However, in practice, you should allow for at least 25% more space than this algorithm indicates, so the capacity of the default log files is around 180 000 job stream instances.

If JnextPlan has neared or exceeded that level, you must make more log space available to DB2®.

In addition to performing the above calculation, you can also determine the log space actually used by a specific instance of JnextPlan and base your log size requirement on that figure.

Determining actual DB2® log file usage

The following is the procedure to verify how much space was used by a successful instance of the JnextPlan command:

1. After JnextPlan has run, log on to the computer where the IBM Workload Scheduler DB2® server is installed, as the DB2® instance owner (UNIX™) or DB2® Administrator (Windows™).
2. Open a DB2® command line window or shell, as follows:

UNIX™

Follow these steps:

- a. Issue the command `su - db2inst1`, or change to the subdirectory `sqllib` of the home directory of the owner of the DB2® instance (by default `db2inst1`)
- b. Launch the command `./db2profile`

Windows™

Select from the **Start** menu, **Programs** → **IBM DB2** → **Command Line Tools** → **Command Window**

3. Run the following command:

```
db2 "get snapshot for database on TWS" > snapdb.txt
```

where "TWS" must be changed to the actual database name if different

4. Open the `snapdb.txt` file and look for a section like this:

```
Log space available to the database (Bytes)= 244315359
Log space used by the database (Bytes)      = 484641
Maximum secondary log space used (Bytes)    = 0
Maximum total log space used (Bytes)       = 581636
Secondary logs allocated currently         = 0
```

The value shown in "Maximum total log space used" is the actual space used for the DB2® logs. This space should be allocated to DB2® using primary log files only: therefore, you should change the number of primary log files and their size as necessary to meet this requirement as a minimum.

In addition, you are recommended to allocate a secondary log space to DB2®. A good choice for the secondary log files is half the number allocated for the primary files.

The snapshot command described in [step 3 on page 474](#) can be run at any time to keep track of the current usage of the DB2® log space, without a noticeable impact on performance. All metrics shown are useful to monitor the current allocation of DB2® primary and secondary logs at any time, and to determine any required changes.

Procedure for changing the maximum DB2® log capacity

Do this as follows:

1. Log on to the computer where the IBM Workload Scheduler DB2® server is installed, as the DB2® instance owner (UNIX™) or DB2® Administrator (Windows™).
2. Open a DB2® command line window or shell, as follows:

UNIX™

Follow these steps:

- a. Issue the command `su - db2inst1`, or change to the subdirectory `sqllib` of the home directory of the owner of the DB2® instance (by default `db2inst1`)
- b. Launch the command `./db2profile`

Windows™

Select from the **Start** menu, **Programs** → **IBM DB2** → **Command Line Tools** → **Command Window**

3. Run the following commands:

```
db2 update db cfg for <database_name> using LOGFILSIZ <log_file_size>
db2 update db cfg for <database_name> using LOGPRIMARY <primary_log_files>
db2 update db cfg for <database_name> using LOGSECOND <secondary_log_files>
```

where:

<database_name>

The name of the database:

- If you are running this from the computer where the DB2® server is installed, the installed default name is *TWS*. Supply this value unless you have changed it.
- You are not recommended to run this procedure from the computer where the DB2® client is installed, but if you do so, the installed default name is *TWS_DB*. Supply this value unless you have changed it.

<log_file_size>

The log file size in 4 KB pages. The default is 1000 (hence the default log file size of 4MB). Look in the DB2® documentation for details of the implications of choosing a larger or a smaller log file size. The maximum value is 262 144 (making the maximum log file size about 1 GB).

<primary_log_files>

The number of primary log files. The default is 40. The total maximum number of log files that DB2® can handle (primary and secondary) is 256. Thus, there is a maximum limit of 256 GB for the log, or approximately 256 million Job Scheduler instances! (maximum 256 files x 1 GB maximum file size)

<secondary_log_files>

The number of secondary log files. The default is 20. If there is enough free space on the file system, these additional log files are dynamically allocated by DB2® as needed (with a small impact on the performance of JnextPlan). Because these are only created if required, it is preferable to increase the

number of secondary files, rather than the primary files. Typically, you allocate 50% of the primary log file value.

In making the calculation to allocate the log files, allow at least 25% more space than you think you require, to avoid that any slight miscalculation causes JnextPlan to fail.

Example: if you have determined from the procedure described in [Determining actual DB2 log file usage on page 474](#) that JnextPlan has a current use of 320 MB, you could calculate as follows:

- a. Increase 320 MB by 25%, giving 400 MB
 - b. Determine if you want more log files, or bigger log files, or both, by reference to the DB2® documentation. For example, you could choose to allocate 40 files with a size of 10 MB, 80 files with a size of 5 MB, or 100 files with a size of 4 MB. For the sake of this example, assume you have chosen 80 files with a size of 5 MB, so your LOGPRIMARY value will be 80.
 - c. Determine the log file size in 4 KB pages to give a log file size of 5 MB - your LOGFILSIZ value will thus be 1250.
 - d. Determine how many secondary log files are required. If you follow the 50% guideline you will need a LOGSECOND value of 40.
4. Log on to the computer where IBM Workload Scheduler is installed as the following user:

UNIX™

root

Windows™

Any user in the *Administrators* group.

5. Access the directory: <TWS_INSTALLATION_PATH>\server_<wouser>\appservertools
6. Stop the WebSphere Application Server Liberty Base using the conman stopappserver command (see [Starting and stopping the application server and appservman on page 436](#))
7. On the computer where the DB2® server is installed, stop and start DB2®, as follows:
 - a. Ensure that no other applications are using this instance of DB2®, or if they are that they can be stopped.
 - b. Issue the following command:

```
db2stop
```

- c. Issue the following command:

```
db2start
```



Note: It is strongly recommended that you stop and start DB2®. If this is a problem for you, you must at least disconnect all applications from the DB2® instance and reconnect them. DB2® will apply the new parameters when you reconnect. If necessary, use the following command to force the disconnection of all open connections:

```
db2 "force application all"
```

8. Start the WebSphere Application Server Liberty Base using the conman startappserver command (see [Starting and stopping the application server and appservman on page 436](#))

Oracle tablespace size

Oracle (RDBMS) is divided in tablespaces which are an allocation of space where datafiles are stored.

The number of transactions that can be managed by the Oracle log files depends on the way the Oracle database is configured, thus it is important to allocate an appropriate table size to avoid issues on performance.

For example, considering a workload of 500000 jobs per day, 80GB of .dbf file size is recommended.

Multiple Dynamic Workload Console production plan reports

From the Dynamic Workload Console you can launch production plan reports. These are heavy users of CPU time, and if they are requested for the entire plan, they can also take some considerable time to produce. If several are running at once, they can have a noticeable impact on the performance of the master domain manager.

If you notice a degradation of performance, you can determine if there are any reports running by checking for the report work files, as follows;

1. Navigate to the operating system's temporary directory
2. Look for files that have the following file name template:

```
TWS-sequential_number-extr
```

Each report currently in progress has one of these work files open. The files are removed when the report is completed.

3. Check the dates of these files, and consider only recent files (if a report fails during production at any time, its file remains in the temporary directory until the next reboot of the master domain manager or you run an operating system cleanup process that discards all files in the temporary directory).

There is no direct action to take, as you must wait until the report completes for the performance to recover.

However, if you note that large numbers of reports are being issued, it might indicate the following scenario:

1. A user issues a report request, expecting it to be available immediately
2. When the report does not appear immediately, the user thinks it has hung, closes and reopens the browser, and reissues the report. The closing of the browser does not stop the report production.
3. The user might repeat this action several times.

In this case, you can take action to remind the user that the production of large reports can be time-consuming, and that it always better to wait.

Dynamic Workload Console - adjusting session timeout settings

About this task

The value assigned to the session timeout settings defines after how many minutes a user is automatically logged out from the WebSphere Application Server Liberty Base. If you plan to perform long running operations, or to have many users

connected concurrently to the Dynamic Workload Console, or expect to have low performance on the system where the Dynamic Workload Console is installed, you might want to edit these values: `httpSession invalidationTimeout="5h"` and `ltpa expiration="1440"`.

Perform these steps to change the values assigned to the timeout settings:

1. Stop WebSphere Application Server Liberty Base:

UNIX™

```
./stopAppServer.sh [-direct]
```

Windows™

```
stopAppServer.bat [-direct]
```

For more information about stopping WebSphere Application Server Liberty Base, see [Application server - starting and stopping on page 432](#).

2. Create a .xml file with this content (i.e. timeout_config.xml):

```
<server description="http_timeout_config">
  <httpSession invalidationTimeout="5h" invalidateOnUnauthorizedSessionRequestException="false"/>
  <ltpa expiration="1440"/>
</server>
```

3. Save the file in the following path: `<DATA_DIR>/usr/dwcServer/configDropins/overrides`
4. Start WebSphere Application Server Liberty Base:

UNIX™

```
./startAppServer.sh [-direct]
```

Windows™

```
startAppServer.bat [-direct]
```



Note: The desired time must be indicated in minutes

For more information, please refer to the WebSphere Application Server Liberty Base documentation at the following links https://www.ibm.com/support/knowledgecenter/en/SSEQTP_liberty/com.ibm.websphere.liberty.autogen.base.doc/ae/rwlp_config_httpSession.html. and https://www.ibm.com/support/knowledgecenter/en/SSEQTP_liberty/com.ibm.websphere.liberty.autogen.base.doc/ae/rwlp_config_ltpa.html

Dynamic Workload Console - Increasing application server heap size

Follow this procedure to increase the Java™ heap size:

1. Log on to the computer where Dynamic Workload Console is installed as the following user:

Windows™ operating systems:

Any user in the *Administrators* group.

UNIX™ operating systems:

root

2. Stop the WebSphere Application Server Liberty Base by running:

Windows™ operating systems:

DWC_home\appservertools\stopAppServer.bat

UNIX™ operating systems:

DWC_home/appservertools/stopAppServer.sh

3. Open the following file:

Windows operating systems:

DWC_DATA_dir\usr\servers\dwcServer\configDropins\overrides\jvm.options

UNIX operating systems:

DWC_DATA_dir/usr/servers/dwcServer/configDropins/overrides/jvm.options

4. Here is an example:

```
-Xms4096m
-Xmx4096m
-Xgcpolicy:gencon
#nursery mem size
-Xmn1024m
```



Note: In case of high workload (more than 50 concurrent users) use 6144 as heap size and 1536 as nursery mem size. The above suggested settings must be applied when the RAM configuration value twice the value of the heap size.

5. Save the file `jvm.option`
6. Start the WebSphere Application Server Liberty Base, by running

Windows operating systems:

DWC_home\appservertools\startAppServer.bat

UNIX operating systems:

DWC_home/appservertools/startAppServer.sh

Dynamic Workload Console graphical views

When viewing graphical views in a supported web browser, especially when there are hundreds of objects, including dependencies, it is recommended that you use either Google Chrome or Mozilla Firefox to guarantee the best possible performance. This applies to the following graphical views:

- Job Stream Graphical View (both model and plan)
- Plan View
- Preproduction Plan View

Chapter 11. Availability

This section describes factors that might affect the availability of IBM Workload Scheduler on a workstation. It covers the following topics:

- [Resolving user ID account on Windows operating systems on page 481](#)
- [Using a temporary directory on UNIX on page 482](#)

Resolving user ID account on Windows® operating systems

About this task

IBM Workload Scheduler needs to resolve the user ID account on Windows® operating systems to verify the security information.

Windows® users can be classified as domain users or local users. Domain users are defined in the domain controller, while local users are defined in the workstations of the network.

For a domain user, IBM Workload Scheduler requests the primary domain controller (or any domain controller for Windows® 2000 or 2003 Active Directory), to identify an available domain controller. It then uses this domain controller identity to type out the structure for the user.

For a local user, IBM Workload Scheduler makes a request to the local workstation. Generally, IBM Workload Scheduler specifies two cases: one for the IBM Workload Scheduler user and one for the streamlogon user.

The following is a list of steps that IBM Workload Scheduler performs to authenticate Windows® users, and the APIs involved:

1. IBM Workload Scheduler looks up the user in the reference domain. For the domain user, the reference domain is the name of the Windows® network. For the local user, it is the name of the local workstation.

API: `LookupAccountName`.

2. If the user is a domain user, IBM Workload Scheduler asks the primary domain controller for any domain controller that is available to resolve the account for the user in the reference domain.

API: `NetGetAnyDCName` for Windows® or `DsGetDcName` for Windows® 2000 or 2003.

3. IBM Workload Scheduler requests the domain controller (or the local workstation if the user is local) for information about the user.

API: NetUserGetInfo.



Note: On Windows® 2000 and 2003, the permissions for this API are contained in the `BUILTIN\ "Pre-Windows 2000 compatible access" group`.

Using a temporary directory on UNIX™

When performing IBM Workload Scheduler operations on UNIX™, temporary files are written to the temporary directory on the local workstation. Ensure that the `<TWS_user>` running operations has *read* and *write* access to this directory.

Chapter 12. License Management in IBM License Metric Tool

According to your IBM Workload Scheduler license, IBM® License Metric Tool helps you maintain your license compliance. By using License Metric Tool, you can generate reports that summarize your license consumption. The generated reports are maintained on the License Metric Tool server and should be periodically reviewed and signed, creating a history for audit purposes in the process. If you are contacted by a third-party software compliance auditor who plans to visit your enterprise to carry out a software audit, ensure that all reports are up-to-date and signed, and then supply copies of reports that cover the time periods that the auditor requests.

The following IBM Workload Scheduler license models are available to customers:

- [Processor Value Unit license model on page 483](#)
- [Per Job license model on page 487](#)
- [Using per job queries when upgrading from a version earlier than 9.4 Fix Pack 2 on page 493](#)

To install and configure IBM® License Metric Tool, see the product documentation License Metric Tool V9.2.0 in IBM Knowledge Center: <https://www.ibm.com/docs/en/license-metric-tool>.

Processor Value Unit license model

About this task

IBM® License Metric Tool generates reports to help you maintain compliance with your Processor Value Unit (PVU) sub-capacity license terms. License Metric Tool can calculate PVU consumption only when **software identification tags** exist and are activated by the customer. The optman global option **licenseType** must be set to **perServer** (default value) to activate **Processor Value Unit** consumption tracking. You can also set the optman global option **licenseType** to **byWorkstation** to specify that the licensing type (either **perServer** or **perJob**) is specified at creation time for each workstation.

License Metric Tool automatically detects the following IBM Workload Scheduler 9.5 chargeable components that are part of the product:

Table 86. Chargeable software components automatically detected by License Metric Tool

Chargeable Software Components
IBM Workload Scheduler agent V9.5
IBM Workload Scheduler agent for z/OS V9.5

To detect and count remote nodes that are managed by IBM Workload Scheduler chargeable components, you must manually deploy software tags because no product code is present on those nodes. License Metric Tool generates and assigns dedicated software tags during the creation of the Readiness Package for the latest release of IBM Workload Scheduler.

The following IBM Workload Scheduler Version 9.5 chargeable components require manual deployment of software tags:

Table 87. Chargeable software components that require software tag deployment on managed nodes

Chargeable Software Components	Assigned Software tags
IBM Workload Scheduler agent-less9.5	ibm.com_IBM_Workload_Scheduler_agent-less-9.5.swidtag
IBM Workload Scheduler for third-party Applications 9.5	ibm.com_IBM_Workload_Scheduler_for_Third_Party_Applications-9.5.swidtag
IBM Workload Scheduler for IBM Applications 9.5	ibm.com_IBM_Workload_Scheduler_for_IBM_Applications-9.5.swidtag



Note: For information about how to match chargeable software components with the IBM Workload Scheduler access methods and application plug-ins, see table [Table 88: IBM Workload Scheduler chargeable access methods and application plug-ins on page 486](#)

Complete the following procedure to manually deploy dedicated software tags **on each managed node** and calculate PVU consumption:

1. Identify the remote nodes that are managed by each of your chargeable components. For example, the remote nodes that are managed by the application server that you are connecting to, with IBM Workload Scheduler plug-in for IBM InfoSphere DataStage.
2. Extract assigned software tags from the `ILMT_IWS_for_Applications_and_agentless.zip` file that is included in your Agent installation media.
3. Place the assigned software tag anywhere on the managed node.
4. Wait for the next software scan in License Metric Tool to have the chargeable software components reported.
5. It is recommended that you check whether License Metric Tool reporting matches with currently managed nodes before signing each License Metric Tool report.

The master domain manager centrally maintains the history of the plug-in jobs that you run in your environment. The history can be used:

- During audits to verify which systems are actually managed by IBM Workload Scheduler.
- To check periodically the result of your License Metric Tool reporting.
- To verify your PVU license entitlement.

An SQL query is provided to access the history in the database. You can run the query either from the command-line interface of your database or by creating your custom SQL report tasks from the Dynamic Workload Console as described in *Dynamic Workload Console User's Guide*.

For each job definition in the database, the SQL query returns the:

1. Type of the plug-in job.
2. Name of the workstation on which the job is defined.

3. Name of the job.
 4. XML file containing the name of the remote server on which the job ran. If the XML file does not contain the name of the remote server, you can find it in the plug-in properties file in the `TWS/JavaExt/cfg` agent folder.
- For DB2, IDS, MSSQL database types:

```
SELECT JOD_TASK_TYPE as Job_type, WF.FOL_PATH as Workstation_folder_name,
WKC_NAME as Workstation_name, JF.FOL_PATH as Job_folder_name, JOD_NAME as
Job_name,
JOD_TASK_STRING as Job_definition
FROM (((MDL.JOD_JOB_DEFINITIONS J inner join MDL.FOL_FOLDERS JF on
J.FOL_ID=JF.FOL_ID)
inner join MDL.WKC_WORKSTATION_CLASSES W on J.WKC_ID=W.WKC_ID)
inner join MDL.FOL_FOLDERS WF on W.FOL_ID=WF.FOL_ID)
WHERE JOD_BY_JSDDL='Y' and UPPER(JOD_TASK_TYPE) NOT IN ('EXECUTABLE',
'DISTRIBUTEDSHADOWJOB', 'ZSHADOWJOB')
ORDER BY JOD_TASK_TYPE, WKC_NAME, JOD_NAME
```

- For Oracle database type:

```
SELECT JOD_TASK_TYPE as Job_type, WF.FOL_PATH as Workstation_folder_name,
WKC_NAME as Workstation_name, JF.FOL_PATH as Job_folder_name, JOD_NAME as
Job_name,
JOD_TASK_STRING as Job_definition
FROM (((twuser.JOD_JOB_DEFINITIONS J inner join twuser.FOL_FOLDERS JF on J.FOL_ID=JF.FOL_ID)
inner join twuser.WKC_WORKSTATION_CLASSES W on J.WKC_ID=W.WKC_ID)
inner join twuser.FOL_FOLDERS WF on W.FOL_ID=WF.FOL_ID)
WHERE JOD_BY_JSDDL='Y' and UPPER(JOD_TASK_TYPE) NOT IN ('EXECUTABLE',
'DISTRIBUTEDSHADOWJOB', 'ZSHADOWJOB')
ORDER BY JOD_TASK_TYPE, WKC_NAME, JOD_NAME
```

where **twuser** is the name of the IBM Workload Scheduler schema.

The following example shows the query output for a **Datastage** job type:

```
datastage NY_1 DS_JOB <?xml version="1.0" encoding="UTF-8"?>
<jSDL:jobDefinition xmlns:jSDL="http://www.ibm.com/xmlns/prod/scheduling/1.0/jSDL"
xmlns:jSDLdatastage="http://www.ibm.com/xmlns/prod/scheduling/1.0/jSDLdatastage"
name="DATASTAGE">
  <jSDL:application name="datastage">
    <jSDLdatastage:datastage>
      <jSDLdatastage:DataStageParameters>
        <jSDLdatastage:DataStagePanel>
          <jSDLdatastage:Logon>
            <jSDLdatastage:Domain>ncxx4175.romelab.it.ibm.com:9080</jSDLdatastage:Domain>
            <jSDLdatastage:Server>ncxx4175</jSDLdatastage:Server>
            <jSDLdatastage:UserName>isadmin</jSDLdatastage:UserName>
            <jSDLdatastage:password>{aes}ScWNLDaHuN9X5sbtvAVky3RVd7gOkJqNerDbFbpwrDg=
          </jSDLdatastage:password>
        </jSDLdatastage:Logon>
      </jSDLdatastage:DataStageParameters>
    </jSDLdatastage:datastage>
  </jSDL:application>
</jSDL:jobDefinition>
```

```

<jsdldatastage:FileRemotePath/>
</jsdldatastage:JobDefinitionGroup>
<jsdldatastage:JobExecutionGroup/>
</jsdldatastage:DataStagePanel>
<jsdldatastage:OptionsPanel>
<jsdldatastage:JobOptionsGroup>
<jsdldatastage:WarningLimitButtonGroup>
<jsdldatastage:NoWarningLimitButton/>
</jsdldatastage:WarningLimitButtonGroup>
<jsdldatastage:RowLimitButtonGroup>
<jsdldatastage:NoRowLimitButton/>
</jsdldatastage:RowLimitButtonGroup>
<jsdldatastage:OperationalMetadataGroup>
<jsdldatastage:UseDefault/>
</jsdldatastage:OperationalMetadataGroup>
</jsdldatastage:JobOptionsGroup>
</jsdldatastage:OptionsPanel>
</jsdldatastage:DataStageParameters>
</jsdldatastage:datastage>
</jsdl:application>
</jsdl:jobDefinition>

```

Table 88. IBM Workload Scheduler chargeable access methods and application plug-ins

IBM Workload Scheduler access methods and application plug-ins	Chargeable Software Components
Remote Command	IBM Workload Scheduler agent-less V9.5
Unixssh	
SAP	IBM Workload Scheduler for third-party Applications V9.5
SAP PI Channel	
SAP BusinessObjects BI	
PeopleSoft	
Oracle E-Business Suite	
Informatica PowerCenter	
Salesforce	
Hadoop Map Reduce	
Hadoop Distributed File System	
Apache Oozie	
Apache Spark	
Amazon EC2	
Microsoft Azure	
IBM Sterling Connect:Direct	IBM Workload Scheduler for IBM Applications V9.5
IBM WebSphere® MQ	

Table 88. IBM Workload Scheduler chargeable access methods and application plug-ins

(continued)

IBM Workload Scheduler access methods and application plug-ins	Chargeable Software Components
IBM InfoSphere DataStage	
IBM® Cognos	
IBM® BigInsights	
IBM® Cloudant	
IBM® SoftLayer	
z/OS	Paid by PVU, manually counted, ILMT not available

Per Job license model

About this task

To generate a report that summarizes your monthly per-job license usage, you can generate a license metric tag file (SLMTag). The SLM tag that is generated applies the **10 monthly jobs** pricing method where, the job count increments by 1 for every 10 successfully executed jobs you run and 1 job is counted when you run anywhere from 1 to 10 jobs. For example, if you run 34 jobs, 4 licenses are counted.

In the **optman** global options, use the **licenseType** keyword to define the pricing model. If you set the **licenseType** keyword to **byWorkstation**, you can then define the pricing model to be applied for each single workstation at creation time, specifying either **perServer** or **perJob**. If you select the **perServer** setting in **optman**, see [Processor Value Unit license model on page 483](#) for more information about tracking license consumption.

The queries listed below apply when you select either the **byWorkstation** or **perJob** pricing models in **optman** to return the license consumption tracking. If you select the **byWorkstation** value, the queries listed below return the number of records with **license type=J** generated by successful jobs on workstations which are set to **license type=perJob**.

You can optionally retrieve consumption information for a subset of workstations. To obtain this information, remove the comment before the lines:

```
-- "((Actual_workstation_name_in_run = 'WKS_NAME1' AND
-- ACTUAL_WKS_FOLDER_NAME_IN_RUN = '/FOL_WKS1/') OR
-- (Actual_workstation_name_in_run = 'WKS_NAME2' AND
-- ACTUAL_WKS_FOLDER_NAME_IN_RUN = '/FOL_WKS2/')) and "
```

remove the double dashes (–), and replace the `'/FOL_WKS1/'`, `'WKS_NAME1'`, `'/FOL_WKS2/'`, `'WKS_NAME2'` strings with your folder and workstation couples.

The master domain manager centrally maintains the history of the jobs that you run in your environment. By using the **optman** global option, **statsHistory**, you can set the number of days for which you maintain the history of the jobs. To track your monthly per-job license usage, set the value of **statsHistory** to 400 (which is the default value). For more information about **statsHistory**, see [Global options - detailed description on page 29](#).

For the SQL statement to generate the SLMTag file, see the following samples:

- For **DB2** database type:

```

SELECT '<SchemaVersion>2.1.1</SchemaVersion>
<SoftwareIdentity>
<Name>IBM Workload Automation</Name>
<PersistentId>3303c35cbc08435080502d621d5cdbff</PersistentId>
<InstanceId>/opt/IBM/TWA</InstanceId>
</SoftwareIdentity>' as xml from sysibm.sysdummy1
UNION
SELECT xml_metrics as xml
FROM (SELECT CONCAT ('<Metric logTime="',CONCAT(current_date,CONCAT
('T',CONCAT(replace(current_time, '.', ':'),CONCAT('+00:00">
<Type>10_MONTHLY_JOBS</Type>
<Value>',CONCAT(JobNbr,CONCAT('</Value>
<Period><StartTime>',CONCAT(Year,CONCAT('-',CONCAT
(trim(VARCHAR_FORMAT(Month,'00')),CONCAT('-01T00:00:01+00:00</StartTime>
<EndTime>',CONCAT(Year,CONCAT('-',CONCAT(trim(VARCHAR_FORMAT(
Month,'00')),CONCAT('-',CONCAT(LAST_DAY,'T23:59:00+00:00</EndTime>
</Period></Metric>'))))))))))))))) as xml_metrics
FROM (SELECT Year, Month,
CASE
when Month = 2 then '28'
when Month = 4 then '30'
when Month = 6 then '30'
when Month = 9 then '30'
when Month = 11 then '30'
else '31'
end as LAST_DAY,
(COUNT(*)+9)/10 AS JobNbr,
current date as current_date,
current time as current_time
from (SELECT unique year(Job_run_date_time) AS Year,
month(Job_run_date_time)AS Month, day(Job_run_date_time) AS day,
JOB_STREAM_WKS_FOL_NAME, JOB_STREAM_WKS_NAME_IN_RUN, JOB_STREAM_FOLDER_NAME, JOB_STREAM_NAME_IN_RUN,
JOB_NAME_IN_RUN FROM MDL.JOB_HISTORY_V
WHERE Job_status='S' and Workstation_license_type='J' and
-- ((Actual_workstation_name_in_run = 'WKS_NAME1' AND
-- ACTUAL_WKS_FOLDER_NAME_IN_RUN = '/FOL_WKS1/') OR
-- (Actual_workstation_name_in_run = 'WKS_NAME2' AND ACTUAL_WKS_FOLDER_NAME_IN_RUN = '/FOL_WKS2/')) and
(Actual_WKS_FOLDER_NAME_IN_RUN, Actual_workstation_name_in_run) not in
(select FOL_PATH, WKS_NAME from MDL.WKS_WORKSTATIONS W JOIN MDL.FOL_FOLDERS
F ON W.FOL_ID=f.FOL_ID where W.WKS_AGENT_TYPE='E'))
GROUP BY Year, Month))
ORDER BY xml desc

```

- For **Oracle** database type:

```

SELECT '<SchemaVersion>2.1.1</SchemaVersion>
<SoftwareIdentity>
<Name>IBM Workload Automation</Name>
<PersistentId>3303c35cbc08435080502d621d5cdbff</PersistentId>
<InstanceId>/opt/IBM/TWA</InstanceId>
</SoftwareIdentity>' as xml from dual
UNION

```



```

SELECT xml_metrics as xml
FROM (SELECT '<Metric logTime="' || cdate || 'T' || ctime || '+00:00">
<Type>10_MONTHLY_JOBS</Type>
<Value>' || JobNbr || '</Value>
<Period>
<StartTime>' || Year || '-' || trim(TO_CHAR(Month,'00')) ||
'-01T00:00:01+00:00</StartTime>
<EndTime>' || Year || '-' || trim(TO_CHAR(Month,'00')) || '-'
|| LAST_DAY || 'T23:59:00+00:00
</EndTime></Period></Metric>' as xml_metrics
FROM (SELECT Year, Month,
CASE
when Month = 2 then '28'
when Month = 4 then '30'
when Month = 6 then '30'
when Month = 9 then '30'
when Month = 11 then '30'
else '31'
end as LAST_DAY,
CAST((COUNT(*)+9)/10 AS INT) AS JobNbr,
TO_CHAR(SYSDATE, 'YYYY-MM-DD') as cdate,
TO_CHAR(SYSDATE, 'HH24:MI:SS') as ctime
from (
SELECT unique EXTRACT(year FROM Job_run_date_time) AS Year,
EXTRACT(month FROM Job_run_date_time) AS Month,
EXTRACT(day FROM Job_run_date_time) AS Day,
JOB_STREAM_WKS_FOL_NAME,
JOB_STREAM_WKS_NAME_IN_RUN,
JOB_STREAM_FOLDER_NAME,
JOB_STREAM_NAME_IN_RUN,
JOB_NAME_IN_RUN
FROM JOB_HISTORY_V
WHERE Job_status='S' and Workstation_license_type='J' and
-- ((Actual_workstation_name_in_run = 'WKS_NAME1' AND
-- ACTUAL_WKS_FOLDER_NAME_IN_RUN = '/FOL_WKS1/') OR
-- (Actual_workstation_name_in_run = 'WKS_NAME2' AND ACTUAL_WKS_FOLDER_NAME_IN_RUN = '/FOL_WKS2/')) and
(Actual_WKS_FOLDER_NAME_IN_RUN, Actual_workstation_name_in_run) not in
(select FOL_PATH, WKS_NAME from WKS_WORKSTATIONS W JOIN FOL_FOLDERS F ON
W.FOL_ID=f.FOL_ID where W.WKS_AGENT_TYPE='E'))
GROUP BY Year, Month))
ORDER BY xml desc

```

- For IDS database type:

```

SELECT '<SchemaVersion>2.1.1</SchemaVersion><SoftwareIdentity><Name>IBM
Workload
Automation</Name><PersistentId>3303c35cbc08435080502d621d5cdbff</Persistent
Id><InstanceId>/opt/IBM/TWA</InstanceId></SoftwareIdentity>' as xml FROM
SYSTABLES
UNION
SELECT xml_metrics as xml FROM (SELECT CONCAT ('<Metric
logTime="' ,CONCAT(current_date,CONCAT('T',CONCAT
(current_time,CONCAT('+00:00"><Type>10_MONTHLY_JOBS</Type><Value>',CONCAT(r
ound(JobNbr,0),CONCAT('</Value><Period><StartTime>',CONCAT(Year,CONCAT ('-
',CONCAT(Month,CONCAT('-01T00:00:01+00:00</StartTime><EndTime>
',CONCAT(Year,CONCAT('- ',CONCAT(Month,CONCAT('- ',CONCAT(LAST_DAY,'T23:59:00+00:00</EndTime>
</Period></Metric>'))))))))))))))) as xml_metrics
FROM (SELECT Year,
replace(TO_CHAR(Month, '**'), '*', '0') AS Month,
CASE
when Month = 2 then '28'
when Month = 4 then '30'
when Month = 6 then '30'

```

```

when Month = 9 then '30'
when Month = 11 then '30'
else '31'
end as LAST_DAY,
(COUNT(*)+9)/10 AS JobNbr,
TO_CHAR(today,'%Y-%m-%d') as current_date,
TO_CHAR(extend (current, hour to second),'%H:%M:%S') as current_time
FROM (SELECT unique year(Job_run_date_time) AS Year,
month(Job_run_date_time) AS Month, day(Job_run_date_time) AS day,
JOB_STREAM_WKS_FOL_NAME, JOB_STREAM_WKS_NAME_IN_RUN,
JOB_STREAM_FOLDER_NAME, JOB_STREAM_NAME_IN_RUN,
JOB_NAME_IN_RUN
FROM MDL.JOB_HISTORY_V
WHERE Job_status='S' and Workstation_license_type='J' and
-- ((Actual_workstation_name_in_run = 'WKS_NAME1' AND
-- ACTUAL_WKS_FOLDER_NAME_IN_RUN = '/FOL_WKS1/') OR
-- (Actual_workstation_name_in_run = 'WKS_NAME2' AND ACTUAL_WKS_FOLDER_NAME_IN_RUN = '/FOL_WKS2/')) and
not exists (select 1 from MDL.WKS_WORKSTATIONS W JOIN MDL.FOL_FOLDERS F ON
W.FOL_ID=F.FOL_ID where W.WKS_AGENT_TYPE='E' AND
Actual_wks_folder_name_in_run = F.FOL_PATH AND
Actual_workstation_name_in_run = W.WKS_NAME))
GROUP BY Year, Month))
ORDER BY xml desc

```

- For **MSSQL** database type:

```

SELECT '<SchemaVersion>2.1.1</SchemaVersion>'
<SoftwareIdentity>
<Name>IBM Workload Automation</Name>
<PersistentId>3303c35cbc08435080502d621d5cdbff</PersistentId>
<InstanceId>/opt/IBM/TWA</InstanceId>
</SoftwareIdentity>' as xml
UNION
SELECT b.xml_metrics as xml
FROM (SELECT '<Metric logTime="' + CONVERT(nvarchar(19), a.datetime, 126)
+ '+00:00">
<Type>10_MONTHLY_JOBS</Type>
<Value>' + CONVERT(varchar(10), a.JobNbr) + '</Value><Period>
<StartTime>' + CONVERT(varchar(10), a.Year) + '-' + RIGHT('00' +
CONVERT(varchar(2),
a.Month), 2) + '-01T00:00:01+00:00</StartTime>
<EndTime>' + CONVERT(varchar(10), a.Year) + '-' + RIGHT('00' + CONVERT(varchar(2),
a.Month), 2) + '-' + a.LAST_DAY + 'T23:59:00+00:00</EndTime>
</Period></Metric>' as xml_metrics
FROM (SELECT Year, Month,
CASE
when Month = 2 then '28'
when Month = 4 then '30'
when Month = 6 then '30'
when Month = 9 then '30'
when Month = 11 then '30'
else '31'
end as LAST_DAY,
(COUNT(*)+9)/10 AS JobNbr, SYSDATETIME() as datetime,
CONVERT (date, SYSDATETIME()) as cdate,
CONVERT (time, SYSDATETIME()) as ctime
FROM (SELECT distinct year(Job_run_date_time) AS Year,
month(Job_run_date_time) AS Month, day(Job_run_date_time) AS day,
Job_stream_wks_fol_name, Job_stream_wks_name_in_run,
Job_stream_folder_name, Job_stream_name_in_run,
Job_name_in_run
FROM MDL.JOB_HISTORY_V
WHERE Job_status='S' and Workstation_license_type='J' and

```

```
--((Actual_workstation_name_in_run = 'WKS_NAME1' AND
-- Actual_wks_folder_name_in_run = '/FOL_WKS1/') OR
-- (Actual_workstation_name_in_run = 'WKS_NAME2' AND Actual_wks_folder_name_in_run = '/FOL_WKS2/')) AND
not exists (select 1 from MDL.WKS_WORKSTATIONS W JOIN MDL.FOL_FOLDERS F ON
W.FOL_ID=F.FOL_ID where W.WKS_AGENT_TYPE='E' AND
Actual_wks_folder_name_in_run = F.FOL_PATH AND
Actual_workstation_name_in_run = W.WKS_NAME)) r
GROUP BY Year, Month) a) b
ORDER BY xml desc
```

The following example shows a license metric tag file with the "**10 monthly number**" of jobs that ran in your environment:

```
<SchemaVersion>2.1.1</SchemaVersion>
<SoftwareIdentity>
<Name>IBM Workload Scheduler</Name>
<PersistentId>3303c35cbc08435080502d621d5cdbff</PersistentId>
<InstanceId>/opt/IBM/TWA</InstanceId>
</SoftwareIdentity>
<Metric logTime="2019-04-09T16:07:20+00:00">
<Type>10_MONTHLY_JOBS</Type>
<Value>2</Value>
<Period><StartTime>2019-03-01T00:00:01+00:00</StartTime>
<EndTime>2019-03-31T23:59:00+00:00</EndTime></Period>
</Metric>
<Metric logTime="2019-04-09T16:07:20+00:00">
<Type>10_MONTHLY_JOBS</Type>
<Value>22</Value>
<Period><StartTime>2019-02-01T00:00:01+00:00</StartTime>
<EndTime>2019-02-28T23:59:00+00:00</EndTime></Period>
</Metric>
```

Queries to verify the number of jobs you run every month

About this task

An SQL query is provided that accesses the job history in the database to verify the number of jobs that you run every month in your environment. The job runs calculated with this query are not grouped in groups of 10 as with the previous queries, but are instead, the total number of jobs that ran.

You can run the SQL query either from the command-line interface of your database, or by creating your custom SQL report tasks from the Dynamic Workload Console, as described in the related section in *Dynamic Workload Console User's Guide*.

- For **DB2** database type:

```
SELECT Year, Month, count(*) AS JobNbr from
(SELECT unique year(Job_run_date_time) AS Year,
month(Job_run_date_time) AS Month, day(Job_run_date_time) AS day,
JOB_STREAM_WKS_FOL_NAME, JOB_STREAM_WKS_NAME_IN_RUN,
JOB_STREAM_FOLDER_NAME, JOB_STREAM_NAME_IN_RUN,
JOB_NAME_IN_RUN
FROM MDL.JOB_HISTORY_V
WHERE Job_status='S' and Workstation_license_type='J' and
--((Actual_workstation_name_in_run = 'WKS_NAME1' AND
-- ACTUAL_WKS_FOLDER_NAME_IN_RUN = '/FOL_WKS1/') OR
```

```
-- (Actual_workstation_name_in_run = 'WKS_NAME2' AND
ACTUAL_WKS_FOLDER_NAME_IN_RUN = '/FOL_WKS2/')) and
(ACTUAL_WKS_FOLDER_NAME_IN_RUN, Actual_workstation_name_in_run) not in
(select FOL_PATH, WKS_NAME from MDL.WKS_WORKSTATIONS W JOIN MDL.FOL_FOLDERS
F ON W.FOL_ID=f.FOL_ID where W.WKS_AGENT_TYPE='E'))
GROUP BY Year, Month
```

- For **ORACLE** database type:

```
SELECT Year, Month, cast (count(*) AS INT) AS JobNbr from
(SELECT unique EXTRACT(year FROM Job_run_date_time) AS Year,
EXTRACT(month FROM Job_run_date_time) AS Month,
EXTRACT(day FROM Job_run_date_time) AS Day,
JOB_STREAM_WKS_FOL_NAME,
JOB_STREAM_WKS_NAME_IN_RUN,
JOB_STREAM_FOLDER_NAME,
JOB_STREAM_NAME_IN_RUN,
JOB_NAME_IN_RUN
FROM JOB_HISTORY_V
WHERE Job_status='S' and Workstation_license_type='J' and
-- ((Actual_workstation_name_in_run = 'WKS_NAME1' AND
-- ACTUAL_WKS_FOLDER_NAME_IN_RUN = '/FOL_WKS1/') OR
-- (Actual_workstation_name_in_run = 'WKS_NAME2' AND
-- ACTUAL_WKS_FOLDER_NAME_IN_RUN = '/FOL_WKS2/')) and
(ACTUAL_WKS_FOLDER_NAME_IN_RUN, Actual_workstation_name_in_run) not in
(select FOL_PATH, WKS_NAME from WKS_WORKSTATIONS W JOIN FOL_FOLDERS F ON
W.FOL_ID=f.FOL_ID where W.WKS_AGENT_TYPE='E'))
GROUP BY Year, Month
```

- For **IDS** database type:

```
SELECT Year, Month, count(*) AS JobNbr from
(SELECT unique year(Job_run_date_time) AS Year, month(Job_run_date_time) AS
Month,
day(Job_run_date_time) AS day, JOB_STREAM_WKS_FOL_NAME,
JOB_STREAM_WKS_NAME_IN_RUN, JOB_STREAM_FOLDER_NAME,
JOB_STREAM_NAME_IN_RUN, JOB_NAME_IN_RUN
FROM MDL.JOB_HISTORY_V
WHERE Job_status='S' and Workstation_license_type='J' and
-- ((Actual_workstation_name_in_run = 'WKS_NAME1' AND
-- ACTUAL_WKS_FOLDER_NAME_IN_RUN = '/FOL_WKS1/') OR
-- (Actual_workstation_name_in_run = 'WKS_NAME2' AND
-- ACTUAL_WKS_FOLDER_NAME_IN_RUN = '/FOL_WKS2/')) and
not exists (select 1 from MDL.WKS_WORKSTATIONS W JOIN MDL.FOL_FOLDERS F ON
W.FOL_ID=F.FOL_ID where W.WKS_AGENT_TYPE='E' AND
Actual_wks_folder_name_in_run = F.FOL_PATH AND
Actual_workstation_name_in_run = W.WKS_NAME))
GROUP BY Year, Month
```

- For **MSSQL** database type:

```
SELECT Year, Month, count(*) AS JobNbr from
(SELECT distinct year(Job_run_date_time) AS Year, month(Job_run_date_time) AS Month,
day(Job_run_date_time) AS day, Job_stream_wks_fol_name,
Job_stream_wks_name_in_run, Job_stream_folder_name,
Job_stream_name_in_run, Job_name_in_run
FROM MDL.JOB_HISTORY_V
WHERE Job_status='S' and Workstation_license_type='J' and
--((Actual_workstation_name_in_run = 'WKS_NAME1' AND
-- Actual_wks_folder_name_in_run = '/FOL_WKS1/') OR
-- (Actual_workstation_name_in_run = 'WKS_NAME2' AND
-- Actual_wks_folder_name_in_run = '/FOL_WKS2/')) AND
not exists (select 1 from MDL.WKS_WORKSTATIONS W JOIN MDL.FOL_FOLDERS F ON
W.FOL_ID=F.FOL_ID where W.WKS_AGENT_TYPE='E' AND
Actual_wks_folder_name_in_run = F.FOL_PATH AND
```

```
Actual_workstation_name_in_run = W.WKS_NAME)) r
GROUP BY Year, Month
```

**Note:**

- All jobs processed or managed by IBM® Workload Scheduler are counted, but the same job counts once if repeated more than once during the same day. To meet this requirement and be considered as the same job, jobs must contain the same *jobstream_workstation_name*, *jobstream_name* and *job_name* strings and not run on a remote engine.
- The SQL queries select only jobs that run successfully. The SQL queries do not count shadow jobs, jobs that run on agent for z/OS, and rerun jobs.

Using per job queries when upgrading from a version earlier than 9.4 Fix Pack 2

About this task

If you are upgrading from a version earlier than 9.4 FP2, then follow the steps listed below before you run the queries listed in [Per Job license model on page 487](#):

1. Upgrade all components in your environment to version 9.4, Fix Pack 2 or later.
2. Depending on your database, run one of the statements listed below as many times as necessary until the **History** table is entirely updated. Several runs might be necessary.
3. Run the queries listed in [Per Job license model on page 487](#).

If you are using a DB2 database, run the following statement:

```
UPDATE ( SELECT * FROM MDL.JHR_JOB_HISTORY_RUNS
WHERE WKC_LICENSE_TYPE = '-' FETCH FIRST 10000 ROWS ONLY )
SET WKC_LICENSE_TYPE = 'J'
```

If you are using an Oracle database, run the following statement:

```
UPDATE JHR_JOB_HISTORY_RUNS
set WKC_LICENSE_TYPE = 'J'
where WKC_LICENSE_TYPE = '-' and rownum <= 10000
```

If you are using an IDS database, run the following statement:

```
UPDATE MDL.JHR_JOB_HISTORY_RUNS SET WKC_LICENSE_TYPE = 'J' WHERE JOB_ID IN
(SELECT JOB_ID FROM
(SELECT FIRST 10000 JOB_ID FROM MDL.JHR_JOB_HISTORY_RUNS WHERE
WKC_LICENSE_TYPE = '-')
)
```

If you are using an MSSQL database, run the following statement:

```
UPDATE TOP (10000) MDL.JHR_JOB_HISTORY_RUNS set WKC_LICENSE_TYPE = 'J'
where WKC_LICENSE_TYPE = '-'
```

Notices

This document provides information about copyright, trademarks, terms and conditions for product documentation.

© Copyright IBM Corporation 1993, 2016 / © Copyright HCL Technologies Limited 2016, 2024

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.

Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. 2016

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.