

HCL Workload Automation
Planning and Installation
Version 10.2.5



Note

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

This edition applies to version 10, release 2, modification level 5 of HCL Workload Automation (program number 5698-T09) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

List of Figures.....	vii
List of Tables.....	viii
About this publication.....	ix
What is new in this release.....	ix
Accessibility	ix
Part I. Planning your HCL Workload Automation environment.....	10
Chapter 1. HCL Workload Automation interfaces.....	14
Chapter 2. Planning the environment.....	16
Distributed workload environment with static scheduling capabilities.....	16
Distributed workload environment with dynamic scheduling capabilities.....	17
Distributed workload environment with static and dynamic scheduling capabilities.....	20
End-to-end workload environment.....	22
Workload environment integrated with external systems.....	22
Distributed-driven workload environment for z/OS®.....	23
Dockerized environment.....	24
Chapter 3. Planning domains.....	26
Single domain network.....	28
Multiple domain network.....	31
Chapter 4. Installation considerations.....	34
Installation paths.....	35
Finding out what has been installed in which HCL Workload Automation instances.....	38
Directories created outside of <i>TWA_home</i> at installation time.....	41
Windows™ services.....	42
Part II. Installing HCL Workload Automation.....	43
Chapter 5. Installing from the command-line interface.....	45
HCL Workload Automation user management.....	49
Windows™ user domain rights and structure.....	50
Considerations for Windows™ domain controllers running Microsoft™ Active Directory.....	51
Enabling product license management.....	52
Typical installation scenario.....	54
Installing Open Liberty	56
Encrypting passwords (optional).....	58
Creating and populating the database.....	59
Creating the HCL Workload Automation administrative user.....	99
Installing the master domain manager and backup master domain manager.....	100
Installing the Dynamic Workload Console servers.....	110
Installing agents.....	113
Installing additional components.....	141
Installing an additional backup domain manager.....	141
Installing dynamic domain components.....	144
Installing the agents on IBM i systems.....	155
Chapter 6. Deploying with containers.....	168
Considerations about deploying with containers.....	169
Deploying with Docker compose.....	170
Prerequisites.....	171
Deploying Docker compose on Linux on Z.....	171
Deploying containers with Docker.....	173
Accessing the Docker containers.....	175
Connecting an on-prem fault tolerant agent to an HCL Workload Automation Server container.....	175
Creating a Docker image to run dynamic agents.....	176
Deploying HCL Workload Automation components using helm charts.....	176
Deploying from Amazon Web Services (AWS) Marketplace.....	176
Getting started.....	177
Creating stacks on AWS CloudFormation.....	178
Accessing the cluster environment and getting credentials.....	179
Downloading packages from the Dynamic Workload Console.....	182
Integrating AI Data Advisor (AIDA).....	182
Upgrading.....	183
Uninstalling.....	184
Deploying on Amazon EKS.....	184
Deploying on Azure AKS.....	185
Deploying on Google GKE.....	185
Workload Automation on HCL SoFy.....	186
HCL Workload Automation on Now Readme File.....	187
Deploying AI Data Advisor.....	190
Troubleshooting.....	192
Container deployment issues.....	192
CURL error 35.....	193
Chapter 7. Post-installation configuration.....	195
Configuring a user registry.....	195
Open Liberty configuration.....	199
Configuring the TLS 1.3 security protocol.....	200
Using SSL for event-driven workload automation (EDWA) behind firewalls.....	203
Configuring your master domain manager and dynamic domain manager in SSL mode.....	203
Part III. Configuring.....	206
Chapter 8. Configuring a master domain manager.....	208

Chapter 9. Configuring a master domain manager configured as backup.....	210
Chapter 10. Configuring a domain manager.....	212
Chapter 11. Configuring a backup domain manager.....	213
Chapter 12. Configuring a dynamic domain manager.....	214
Chapter 13. Configuration steps for a dynamic domain manager configured as backup.....	215
Chapter 14. Configuring a dynamic agent.....	216
Automatically register agents to pools.....	217
Revoking and reissuing a JSON Web Token.....	219
Chapter 15. Configuring a remote command-line client.....	221
Configuring SSL connection between remote command-line client and master domain manager.....	222
Chapter 16. Configuring a z-centric agent on Windows operating systems.....	224
Chapter 17. Adding a feature.....	225
Part IV. Upgrading.....	228
Chapter 18. Downloading installation images on your workstation.....	232
Chapter 19. Upgrading from the CLI.....	233
Before upgrading.....	233
Scanning system prerequisites for HCL Workload Automation.....	236
Connecting the Dynamic Workload Console to a new node or database.....	237
Performing a direct upgrade from v 9.5.0.x or 10.x.x to v 10.2.5.....	238
Upgrading WebSphere Application Server Liberty.....	239
Performing a direct upgrade of the Dynamic Workload Console and its database.....	242
Performing a direct upgrade of the dynamic domain manager, its backups, and their database.....	245
Performing a direct upgrade of the backup master domain manager	246
Performing a direct upgrade of the master domain manager	251
Upgrading agents and domain managers.....	255
Parallel upgrade from version 9.5.0.x or 10.x.x to version 10.2.5.....	257
Converting default certificates.....	258
Upgrading WebSphere Application Server Liberty.....	260
Encrypting passwords (optional).....	262
Upgrading the Dynamic Workload Console and its database.....	264
Creating the HCL Workload Automation administrative user.....	266
Upgrading the database for the server components.....	268
Installing a new dynamic domain manager configured as a backup.....	271
Installing the new master domain manager configured as a backup.....	276
Customizing and submitting the optional FINAL job stream.....	283
Installing a new backup dynamic domain manager.....	285
Cleaning up your environment.....	288
Upgrading agents and domain managers.....	289
Parallel upgrade from version 9.4.0.x to version 10.2.5.....	311
Configuring TLS to the appropriate version.....	312
Converting default certificates.....	313
Installing WebSphere Application Server Liberty.....	315
Encrypting passwords (optional).....	317
Creating and populating the database for the Dynamic Workload Console.....	319
Installing the Dynamic Workload Console.....	334
Creating the HCL Workload Automation administrative user.....	337
Upgrading the database for the server components.....	338
Installing a new dynamic domain manager configured as a backup.....	341
Installing the new master domain manager configured as a backup.....	348
Installing a new backup master domain manager.....	359
Upgrading agents and domain managers.....	366
Enabling product encryption after upgrading.....	368
Enabling API Key authentication after upgrading.....	369
Upgrading when there are corrupt registry files.....	370
Upgrading in a mixed-version environment when using default certificates.....	371
Chapter 20. Updating containers.....	374
Updating containers when using default certificates.....	375
Chapter 21. FAQ - Upgrade procedures.....	378
Part V. Enabling and disabling FIPS.....	379
Chapter 22. Enabling FIPS at installation time.....	380
Chapter 23. Enabling or disabling FIPS at upgrade time.....	382
Chapter 24. Upgrading from a FIPS-enabled environment.....	385
Chapter 25. Enabling or disabling FIPS after installing or upgrading.....	387
Part VI. Moving from on-premises to cloud.....	389
Part VII. Troubleshooting installation, migration, and uninstallation.....	397
Chapter 26. The twsinst log files.....	399
Chapter 27. Analyzing return codes for agent installation, upgrade, restore, and uninstallation.....	400
Chapter 28. Problem scenarios: install, reinstall, upgrade, migrate, and uninstall.....	403

Installation or upgrade fails on RHEL version 9 and later.....	403	Certificates download to agents - AgentCertificateDownloader script.....	482
Installing or linking a fault-tolerant agent earlier than 10.2 in an environment configured with new default or new custom certificates.....	404	Downloading certificates or JWT using a different user.....	487
Dynamic agents not connecting after certificate rotation when using JWT.....	404	Notices.....	cdlxxxix
Uninstallation of Dynamic Workload Console fails....	405	Index.....	493
Error in testing a connection or running reports on an engine returned from Fix Pack 1 to GA level when using an MSSQL database.....	406		
Error in upgrading the HCL Workload Automation database when using a DB2 database.....	406		
Problems in encrypting the useropts file.....	407		
WebSphere Application Server Liberty server does not start when applying a fix pack to the backup master domain manager.....	407		
Error received when creating MSSQL database.....	408		
Incorrect collation settings in PostgreSQL database.....	409		
Chapter 29. Uninstalling HCL Workload Automation manually.....	410		
Uninstalling manually on Windows™ operating systems.....	410		
Uninstalling manually on UNIX™ operating systems.....	412		
Problems during manual uninstall.....	414		
File deletion on Windows™ too slow.....	414		
Part VIII. Uninstalling.....	415		
Chapter 30. Uninstalling the main components.....	416		
Uninstalling a backup master domain manager.....	416		
Uninstalling a master domain manager.....	417		
Uninstalling the Dynamic Workload Console.....	418		
Uninstalling a dynamic domain manager or its backup.....	419		
Uninstalling a dynamic domain manager maintaining a correct hierarchy in the network.....	421		
Uninstalling agents using the twsinst script.....	422		
Uninstalling dynamic and z-centric agents on IBM i systems.....	424		
The twsinst script log files on IBM i systems....	425		
Appendix A. Reference.....	427		
Optional password encryption - secure script.....	427		
Database configuration - configureDb script.....	430		
Server components installation - serverinst script.....	442		
Dynamic Workload Console installation - dwcinst script.....	456		
Agent installation parameters - twsinst script.....	465		
File Proxy installation - fileproxyinst script.....	478		
File Proxy start - fileproxystart script.....	480		
File Proxy stop - fileproxystop script.....	481		
File Proxy uninstallation - uninstall script.....	481		

List of Figures

Figure 1: Graphical overview of a typical HCL Workload Automation environment.....	11
Figure 2: Distributed workload environment with static scheduling capabilities.....	17
Figure 3: Distributed workload environment with dynamic scheduling capabilities.....	19
Figure 4: Distributed workload environment with static and dynamic scheduling capabilities.....	21
Figure 5: Workload environment integrated with external systems.....	23
Figure 6: Distributed-driven workload environment for z/OS®.....	24
Figure 7: Dockerized environment configuration.....	25
Figure 8: Single domain topology.....	29
Figure 9: Single domain topology on multiple sites.....	30
Figure 10: Multiple domain topology.....	32
Figure 11: Typical HCL Workload Automation architecture.....	55
Figure 12: Install fresh Dynamic Workload Console.....	334

List of Tables

Table 1: Features partially or not supported for dynamic scheduling.....	20	Table 26: Valid values for -lang and LANG parameter.....	447
Table 2: Symbolic link options.....	34	Table 27: Valid values for -lang and LANG parameter.....	461
Table 3: Required information.....	101	Table 28: Valid values for -lang and LANG parameter.....	470
Table 4: Valid values for -lang and LANG parameter.....	107	Table 29: Supported combinations and mutual exclusions for authentication and certificate-related parameters.....	476
Table 5: Required information.....	111		
Table 6: Valid values for -lang and LANG parameter.....	124		
Table 7: Supported combinations and mutual exclusions for authentication and certificate-related parameters.....	130		
Table 8: Installation syntax for agent installation with agents in the same network zone.....	136		
Table 9: Installation syntax for agent installation with agents in different network zones.....	138		
Table 10: Required information.....	142		
Table 11: Required information.....	151		
Table 12: Valid values for -lang and LANG parameter.....	162		
Table 13: Open Liberty configuration files.....	200		
Table 14: Required information.....	271		
Table 15: Required information.....	276		
Table 16: Required information.....	285		
Table 17: Required and optional attributes for the definition of a centralized agent update job.....	304		
Table 18: Required information.....	335		
Table 19: Required information.....	342		
Table 20: Required information.....	346		
Table 21: Required information.....	350		
Table 22: Required information.....	360		
Table 23: Windows operating system agent return codes.....	400		
Table 24: UNIX or Linux operating system agent return codes.....	401		
Table 25: Valid values for -lang and LANG parameter.....	433		

About this publication

About this task

This *HCL Workload Automation Planning and Installation* provides information for planning, installing, migrating, and configuring an HCL Workload Automation network.

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 *Overview*, section *Summary of enhancements*.

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 detailed information, see the appendix about accessibility in the *HCL Workload Automation User's Guide and Reference*.

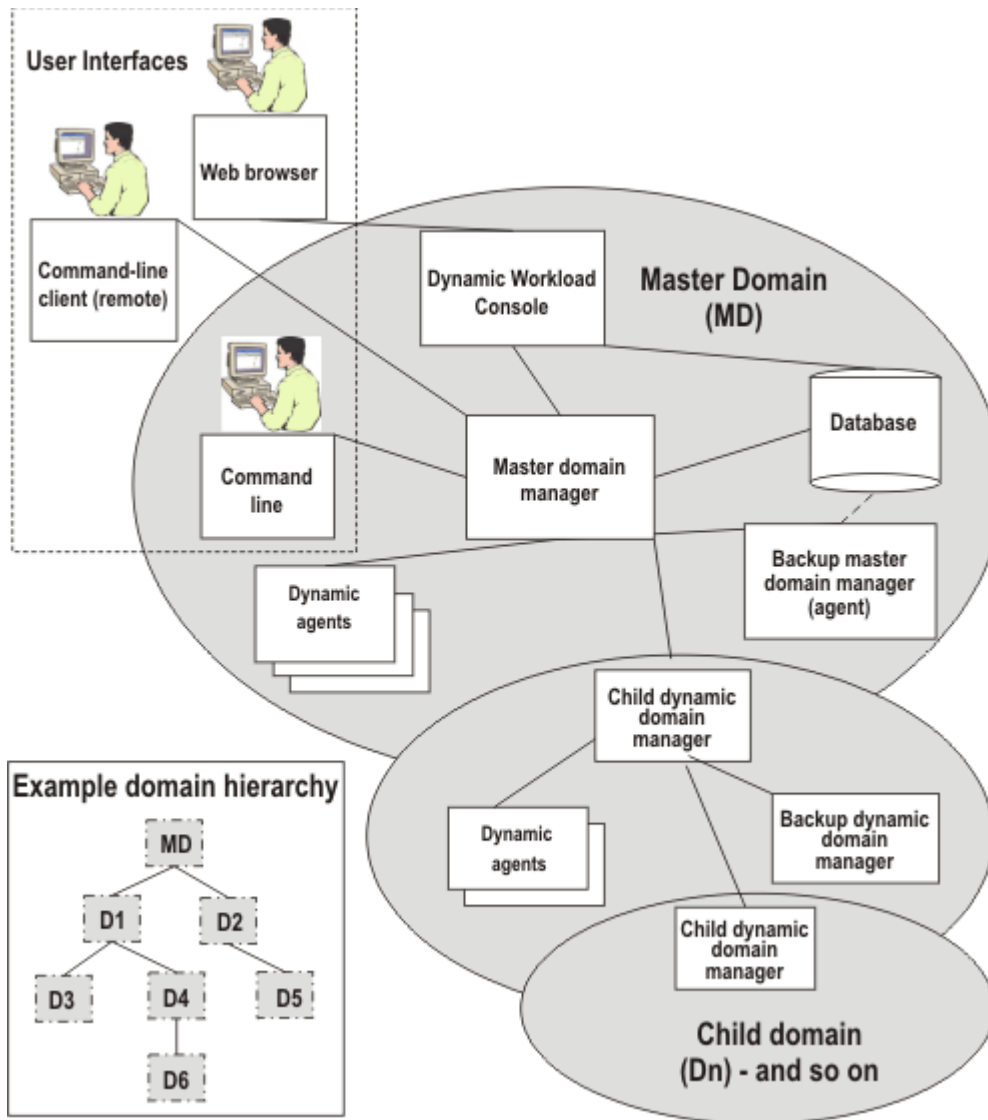
Part I. Planning your HCL Workload Automation environment

HCL Workload Automation orchestrates unattended, scheduled, and event-driven tasks for business and IT processes across on-premises and cloud environments organized in a network. A network consists of a set of linked workstations on which you perform job scheduling and processing to automate and manage your workflows. An HCL Workload Automation network is composed of a master domain manager, one or more Dynamic Workload Console servers, dynamic domain managers, and dynamic agents. You might also have fault-tolerant agents, extended agents, standard agents connected to the master domain manager or to domain managers.

About this task

[Figure 1: Graphical overview of a typical HCL Workload Automation environment on page 11](#) gives a graphical overview of a typical HCL Workload Automation environment:

Figure 1. Graphical overview of a typical HCL Workload Automation environment



In [Figure 1: Graphical overview of a typical HCL Workload Automation environment on page 11](#) the master domain is shown with the main components to run your workload, and two levels of subdomain. The available user interfaces are also indicated. An example is provided of the basic domain hierarchical structure, where each domain is named "D1", "D2, and so on. All of these concepts are explained in the following section.

HCL Workload Automation features the following components:

Master domain manager

The master domain manager is the highest level workstation of an HCL Workload Automation network. It contains or connects to the relational database that stores scheduling object definitions. It creates or updates a production plan when the plan is created or extended and then distributes the plan to the network. It performs all logging and reporting for the network. It can perform the role of event processing server for the event-driven workload automation feature.

Backup master domain manager

Define a backup master domain manager at installation to point to either the database being used by the master domain manager or to a mirror of that database. In this way the backup master domain manager has the latest data available to it at all times and can take over the role of master domain manager seamlessly, in case the master becomes unavailable.

Dynamic domain manager

Install this component if you need a multi-domain network . All domains below the master domain have dynamic domain managers to manage the workstations in their domains. Each dynamic domain manager is an agent in the domain of the next higher level. All communications to and from the dynamic agents in the domain are routed through the dynamic domain manager. To define a dynamic domain manager, install a dynamic domain manager and then perform the [Configuring a dynamic domain manager on page 214](#) procedure.

Backup dynamic domain manager

Install this component if you want a backup to your dynamic domain manager. The backup points to either the database being used by the dynamic domain manager or to a mirror of that database. If your dynamic domain manager experiences problems, you can switch to it with a simple procedure.

Agent

An agent is a workstation in the network that runs the jobs which are controlled by the HCL Workload Automation master domain manager. Several types of agents are available, as follows:

Dynamic agent

An agent that has the following capabilities:

Run workload dynamically

It communicates with the server the status of its resources. In this way the product is able to dynamically run your workload to the best available resources by:

- Automatically discovering scheduling environment resources.
- Automatically following resource changes
- Requesting additional resources when needed
- Matching job requirements to available resources
- Controlling and optimizing use of resources

The characteristics listed above provide high availability and load balancing potentialities to your environment and well suit virtualized environments.

When a job is submitted, either as part of a job stream in the plan or through ad hoc submission, HCL Workload Automation checks the job requirements, the available resources and the related characteristics and submits the job to the resource that best meets the requirements to run it.

Manage dynamic workload broker logical resource

It can remotely run, from the agent, the dynamic workload broker **resource** command on the server. To manage the **resource** command you must also install the Java™ run time.

After installing the agent, you define its type by using [Configuring a dynamic agent on page 216](#).

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. In this way, communication is concentrated in a single connection, reducing the number of connections to the master domain manager or to the dynamic domain manager. For more information about the gateway parameters specified when installing a dynamic agent, see [Agent installation parameters - twsinst script on page 119](#).

For more information about gateway configuration, see the topic about configuring dynamic agent communications through a gateway in *Administration Guide*.

Extended agent

Extended agents are logical definitions (hosted by a physical workstation) used to extend job processing to selected applications (SAP R/3, PeopleSoft, and z/OS®). For information about installing an extended agent, see the topic about installing agents in Planning and Installation .

Fault-tolerant agent

A fault-tolerant agent can resolve local dependencies and launch jobs in the absence of a domain manager. It has a copy of the production control file. This allows fault-tolerant agents to continue processing even if the dynamic domain manager or the network connection is down. With a simple reconfiguration, they can serve as subordinate *domain managers*. To define a fault-tolerant agent, install a fault-tolerant agent on your workstation and then define it as fault-tolerant in the workstation definition.

Standard agent

An agent that launches jobs only under the direction of its domain manager. It is not fault-tolerant. To define a standard agent, install a fault-tolerant agent on your workstation and then define it as a standard agent in the workstation definition.

Chapter 1. HCL Workload Automation interfaces

The HCL Workload Automation features several user interfaces from which you can manage your production environment.

About this task

You can manage your production environment from the following user interfaces:

Master domain manager command lines

The master domain manager command lines are installed automatically when you install the master domain manager. This command lines interface are run only from the workstation serving as the master domain manager. From the command lines, you can administer the master specific binaries and options. A backup master domain manager command lines also exist on the master domain manager configured as backup instance.

Dynamic Workload Console

The web-based interface for creating, modifying, monitoring, controlling, and deleting HCL Workload Automation objects. You can interface with the console from any system in the network where a supported web browser is installed. When you install a Dynamic Workload Console also the **z/OS® Connector** is installed, which is a component that connects HCL Workload Automation for Z and the Dynamic Workload Console. For more information, see *HCL Workload Automation for Z: Planning and Installation Guide*.

Integrations available on Automation Hub

Automation Hub provides an ever-growing number of integrations, software components that enable you to integrate third-party processes into the Dynamic Workload Console and enhance your automation capabilities. A great solution to automate your business workflows and manage all your processes from a single point of control. Check out the full collection at [Automation Hub](#).

Orchestration CLI (OCLI)

Orchestration CLI is a stand-alone command-line application that you can download and install independently without requiring any other HCL Workload Automation component. You can install Orchestration CLI on any workstation where you want to manage and control workflows. It is designed to replace the composer and conman commands, by providing a more modern, efficient, and versatile interface. By using Orchestration CLI, you can automate tasks efficiently, reducing manual effort and operational overhead. Orchestration CLI helps you streamline command-line interactions, enhance cross-platform compatibility, and build a more efficient workload automation process. It also simplifies maintenance, lowers costs, and minimizes IT requirements.

Orchestration CLI also provides a more modern and user-friendly interface, and is designed to be intuitive and efficient, making it easier for administrators and users to complete tasks. It combines modernity, compatibility, and enhanced functionality, when compared to the conman and composer command line.

Command-line client

A component of HCL Workload Automation installed only with a fault-tolerant agent that allows you to implement the following commands on the master domain manager from another workstation: The commands you can use are the following:

- Composer
- Optman
- Planman showinfo and unlock (the other planman commands must be run locally on the master domain manager)

dynamic workload broker command line

Installed and configured automatically when you install a master domain manager. It includes commands to directly submit and manage jobs for dynamic scheduling, manage job JSDL definitions and resources, and more. For more information, see the section about using utility commands in the dynamic environment in *User's Guide and Reference*

Chapter 2. Planning the environment

Typical installation scenarios for products and components.

These typical scenarios for HCL Workload Automation show how to deploy specific solutions on the minimum possible system resources.

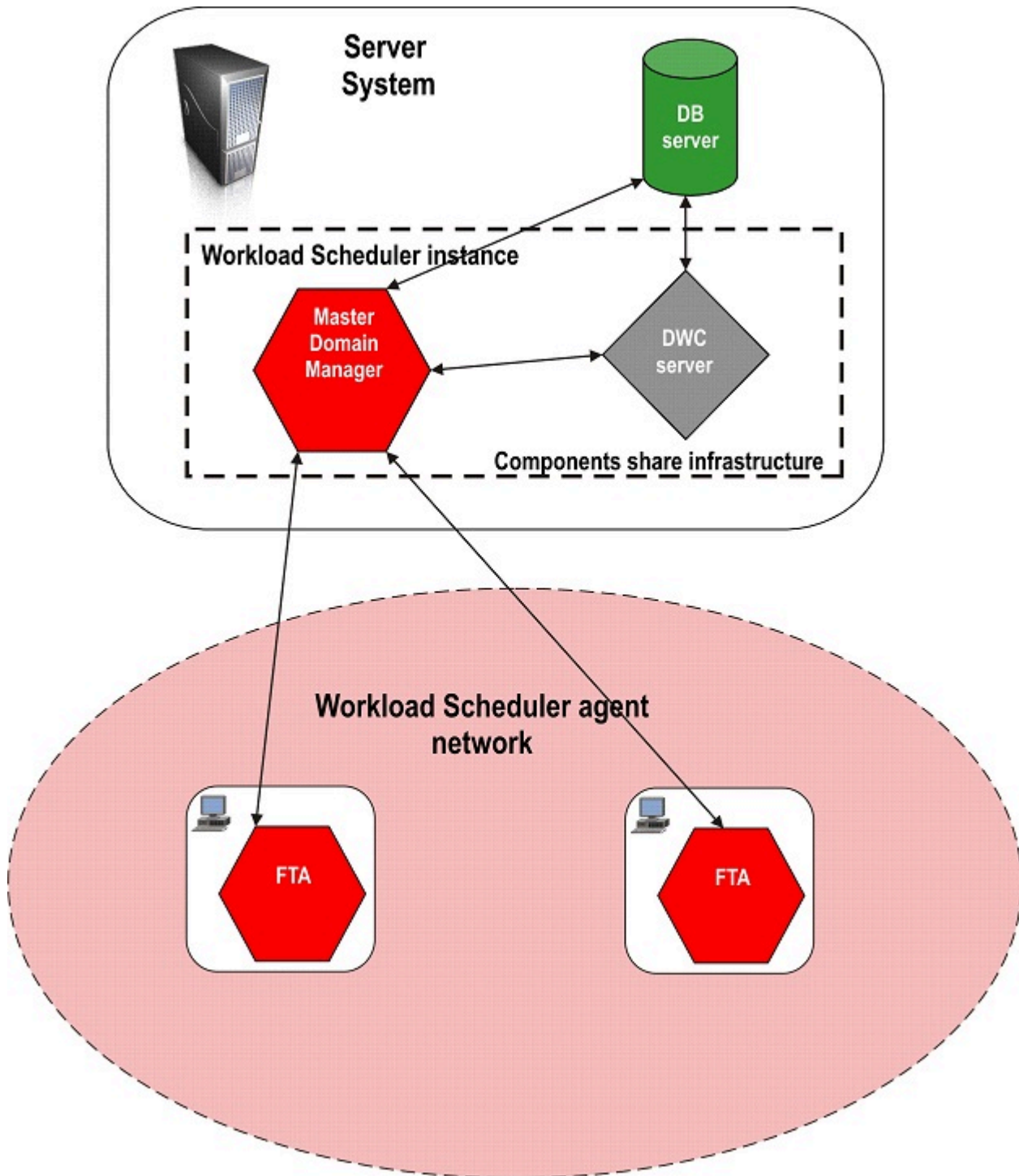
Distributed workload environment with static scheduling capabilities

Configuration to run workload statically across your distributed network.

Use this configuration to run workload statically across your distributed network. [Figure 2: Distributed workload environment with static scheduling capabilities on page 17](#) shows the system resources needed to install a fully-working HCL

Workload Automation environment for managing your distributed workload.

Figure 2. Distributed workload environment with static scheduling capabilities



Distributed workload environment with dynamic scheduling capabilities

Use this configuration to run workload dynamically across your distributed network.

The run time environment is used to:

- Run on the agent job types with advanced options, both those supplied with the product and the additional types implemented through the custom plug-ins.
- Enable the capability to remotely run, from the agent, the dynamic workload broker resource command on the server.

For information about dynamic scheduling, how to run application job plug-ins and the dynamic workload broker resource command on the server, see *HCL Workload Automation: Scheduling Workload Dynamically*.

In this configuration, you can choose whether or not to add the run time environment for Java™ jobs to the agent.

Figure 3: [Distributed workload environment with dynamic scheduling capabilities on page 19](#) shows the system resources required to install a fully working HCL Workload Automation environment for running your distributed workload dynamically.

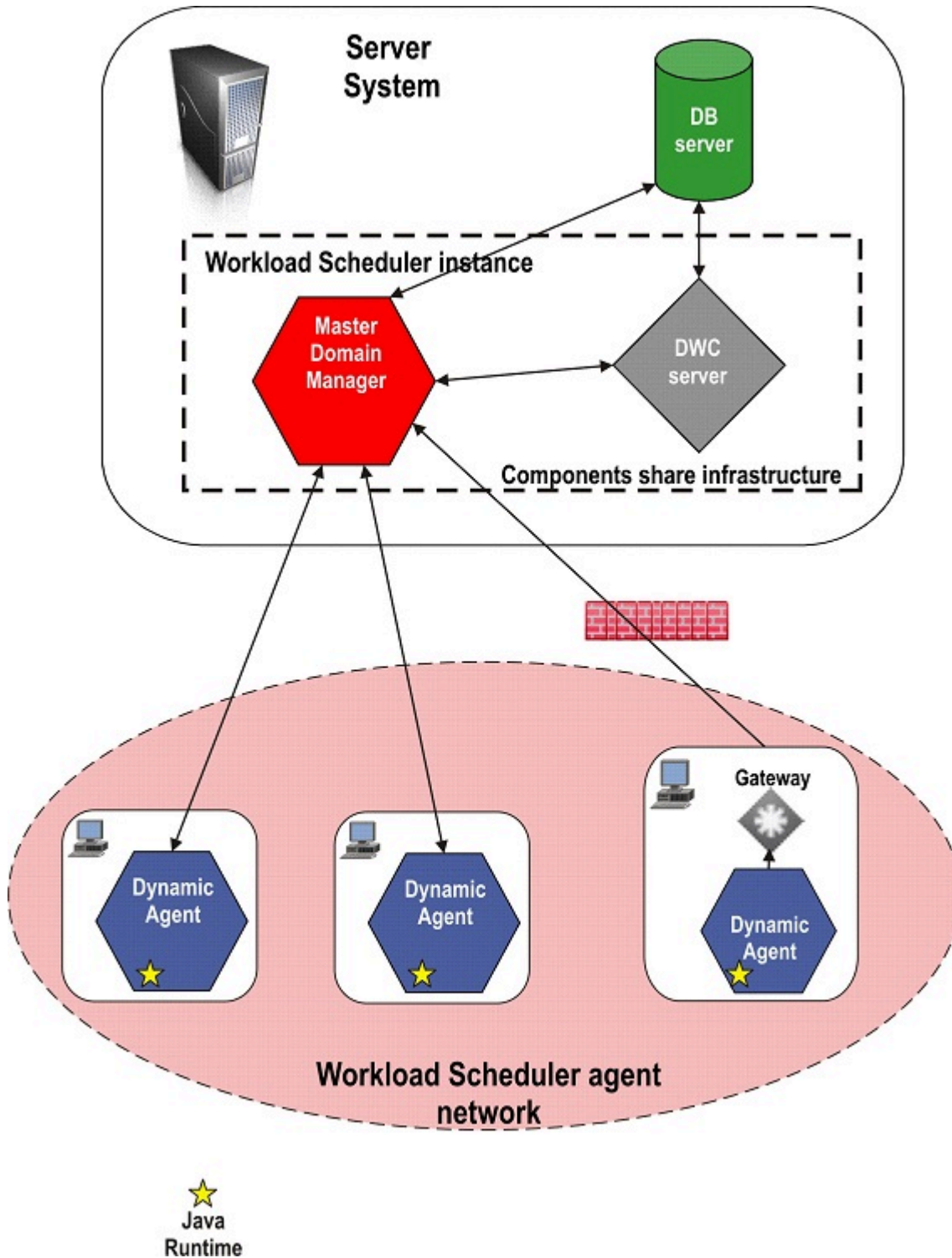


Note: A dynamic agent can be directly connected to its master domain manager or through a dynamic domain manager as shown in [Distributed workload environment with static and dynamic scheduling capabilities on page 20](#). In more complex network topologies where the master domain manager or the dynamic domain manager cannot directly communicate with the dynamic agent, you can configure your dynamic agents to use a local or remote gateway. For more information about the gateway parameters specified when installing a dynamic agent, see [Agent installation parameters - twsinst script on page 119](#). For more information about the gateway parameters specified when installing a dynamic agent, see [Agent installation parameters - twsinst script on page 119](#).




For more information about gateway configuration, see the network communications information in the *Administration Guide*.

Figure 3. Distributed workload environment with dynamic scheduling capabilities



Dynamic scheduling supports most of the HCL Workload Automation features for static scheduling. The [Table 1: Features partially or not supported for dynamic scheduling on page 20](#) lists some features or properties that are partially or not supported.

Table 1. Features partially or not supported for dynamic scheduling

Feature	agent and HCL Workload Automation for Z agent
 Note: For more details about the events type, see <i>HCL Workload Automation User's Guide and Reference: Appendixes - Event-driven workload automation event and action definitions</i>	TivoliWorkloadSchedulerObjectMonitor events supported.
	FileMonitor events supported, except for IBM i systems.
	TivoliWorkloadSchedulerApplicationMonitor events not supported.
Utility commands (datecalc, jobinfo, and so on).	Not supported.

Distributed workload environment with static and dynamic scheduling capabilities

Use this configuration to run workload both statically and dynamically across your distributed network.

The run time environment is used to:

- Run on the agent job types with advanced options, both those supplied with the product and the additional types implemented through the custom plug-ins.
- Enable the capability to remotely run, from the agent, the dynamic workload broker resource command on the server.

For information about dynamic scheduling, how to run application job plug-ins and the dynamic workload broker resource command on the server, see *HCL Workload Automation: Scheduling Workload Dynamically*.

In this configuration, you can choose whether or not to add the run time environment for Java™ jobs to the agent.

[Figure 4: Distributed workload environment with static and dynamic scheduling capabilities on page 21](#) shows the system resources required to install a fully working HCL Workload Automation environment for running your distributed workload both statically and dynamically. HCL Workload Automation requires a fault-tolerant agent and a dynamic agent to be installed on every system where jobs are to be scheduled statically or dynamically.

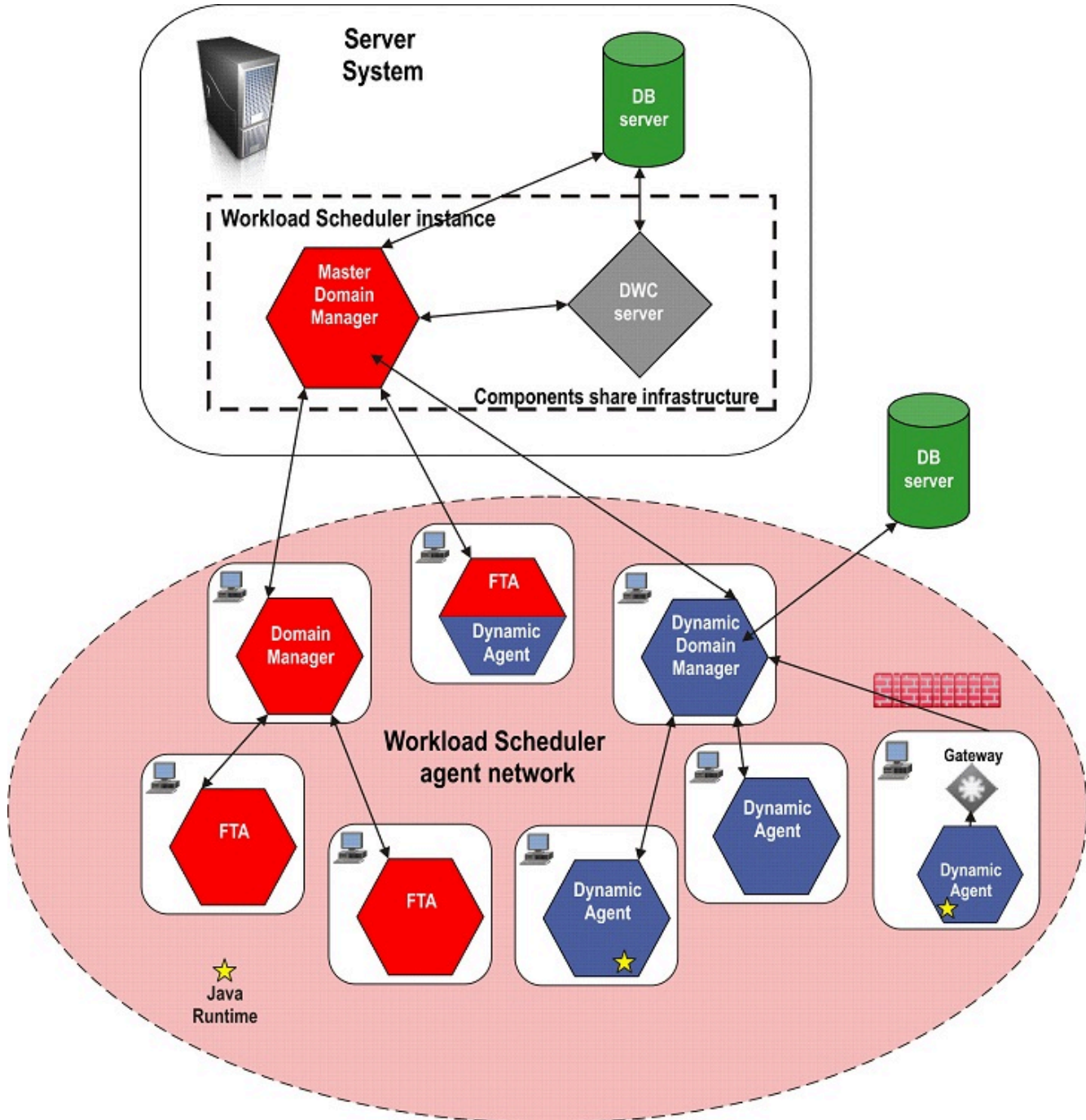


Note: A dynamic agent can be directly connected to its master domain manager or through a dynamic domain manager as shown in [Figure 4: Distributed workload environment with static and dynamic scheduling capabilities on page 21](#). In more complex network topologies where the master domain manager or the dynamic domain manager cannot directly communicate with the dynamic agent, you can configure your dynamic agents to use a local or remote gateway. For more information about the gateway parameters specified when installing a dynamic agent, see [Agent installation parameters - twsinst script on page 119](#).



For more information about gateway configuration, see the network communications information in the *Administration Guide*.

Figure 4. Distributed workload environment with static and dynamic scheduling capabilities



For a list of features partially or not supported in a mixed environment, see [Table 1: Features partially or not supported for dynamic scheduling on page 20](#).

End-to-end workload environment

In an end-to-end workload environment (agent connected to the z/OS® system), you can define different types of configurations.

You can define the following types of configurations:

To run your workload statically:

Using HCL Workload Automation Agents (z-centric)

Use the z-centric end-to-end scheduling environment to schedule and control static workload from the mainframe to distributed systems with a low cost of ownership. On the distributed system, you install HCL Workload Automation Agents and connect them to the z/OS® controller.

For information about how to install the HCL Workload Automation Agent, see *HCL Workload Scheduler for Z: Planning and Installation*. For information about how to use the HCL Workload Automation Agent, see *Scheduling End-to-end with z-centric Capabilities*.

To run your workload dynamically:

Using HCL Workload Automation Agents (z-centric) with dynamic capabilities

Use the z-centric end-to-end scheduling environment to schedule and control dynamic workload from the mainframe to distributed systems with a low cost of ownership. On the distributed system, you install HCL Workload Automation Agents, add dynamic scheduling capabilities and connect them to a dynamic domain manager that must be connected to the z/OS® controller. For information about how to:

- Install a dynamic domain manager see [Installing dynamic domain components on page 144](#).
- Install HCL Workload Automation Agents, see *HCL Workload Scheduler for Z: Planning and Installation*.
- Use HCL Workload Automation Agents, see *Scheduling End-to-end with z-centric Capabilities*.

Workload environment integrated with external systems

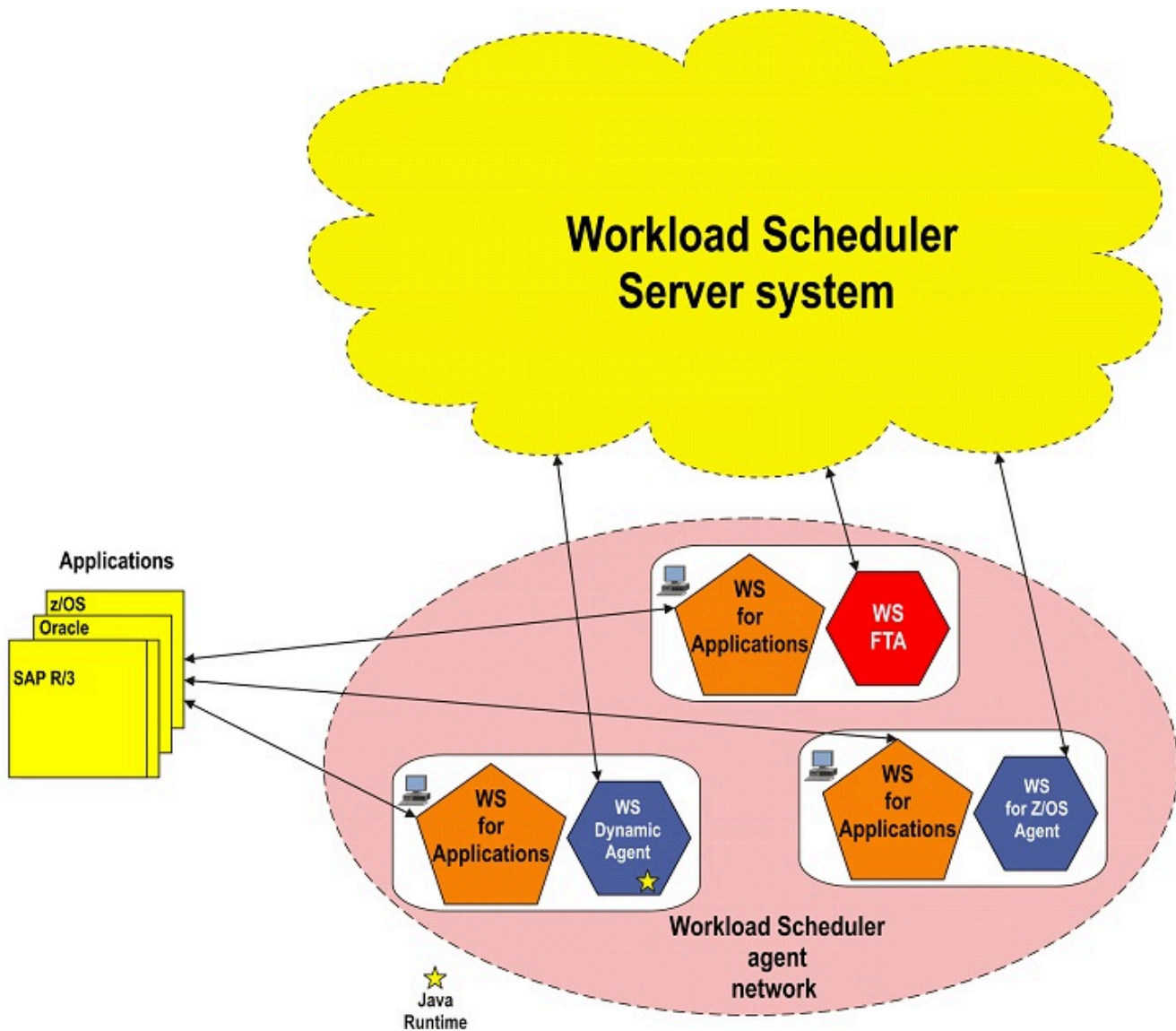
Configuration to extend HCL Workload Automation capabilities for scheduling on external applications.

Use this configuration to extend HCL Workload Automation capabilities for scheduling on external applications, such as SAP and PeopleSoft using HCL Workload Automation.

[Figure 5: Workload environment integrated with external systems on page 23](#) shows a sample environment including the agents needed to extend HCL Workload Automation scheduling capabilities on one or more external applications using HCL Workload Automation. You can install HCL Workload Automation on the master domain manager, on a fault-tolerant agents, on dynamic agents, and on HCL Workload Automation Agents.

For information about HCL Workload Automation, see the *HCL Workload Automation: User's Guide* documentation.

Figure 5. Workload environment integrated with external systems



Note: Installing HCL Workload Automation on an agent (master domain manager, domain manager, fault-tolerant agent, standard agent, dynamic agent, HCL Workload Automation Agent) is the correct deployment scenario in an end-to-end environment.

Distributed-driven workload environment for z/OS®

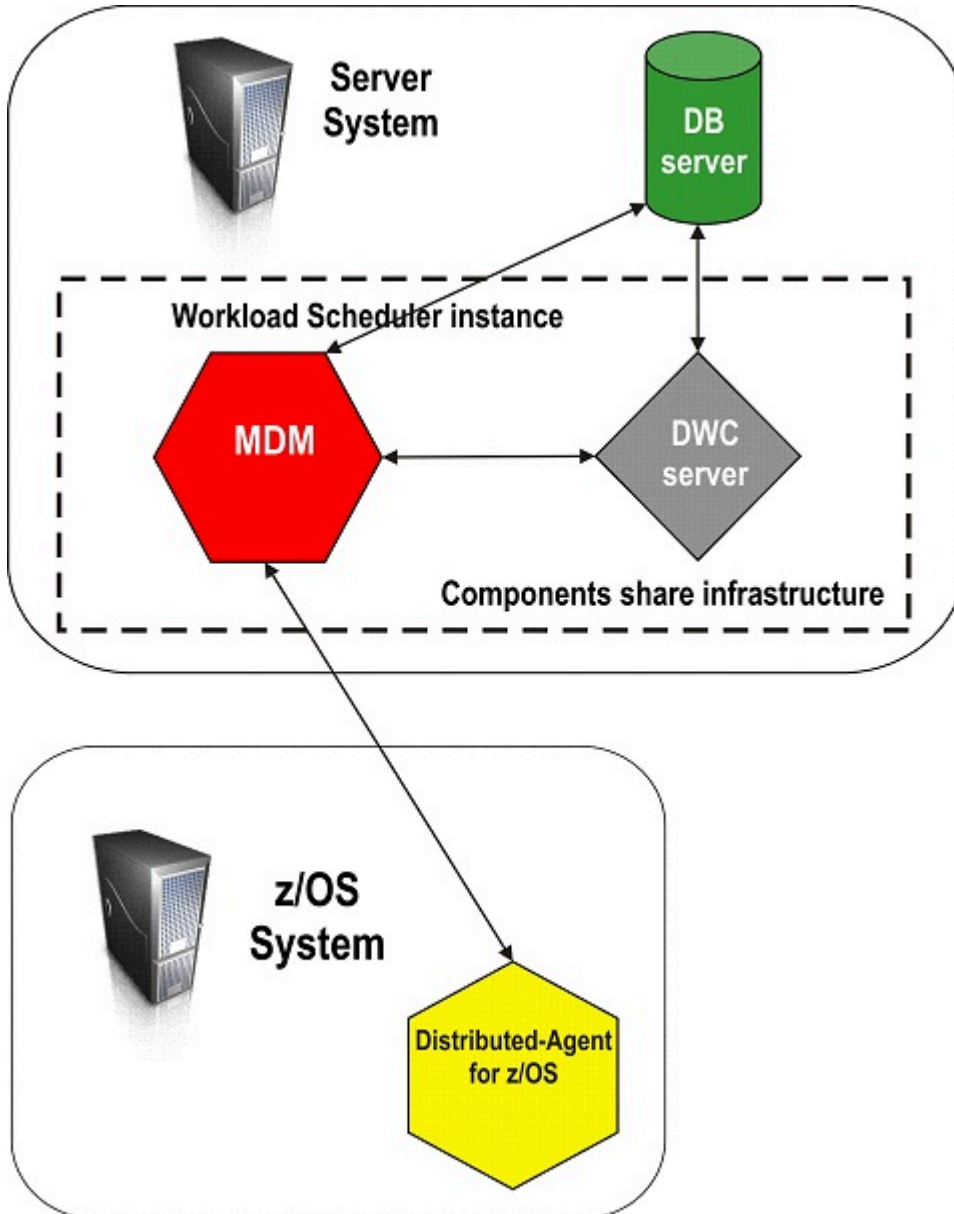
Configuration used when submitting from the HCL Workload Automation.

Use this configuration to submit from the HCL Workload Automation (using the dynamic workload broker component installed with the master domain manager or the dynamic domain manager) workload to be processed by JES2, without having to define the workload on the z/OS® system.

Figure 5: Workload environment integrated with external systems on page 23 shows the minimum system resources needed to install a distributed-driven environment, where the HCL Workload Automation distributed-Agent for z/OS® represents a lightweight end-to-end scheduling solution where you define and manage on the distributed side the workload that is to be processed by JES2.

For information about HCL Workload Automation distributed-Agent for z/OS®, see the *HCL Workload Automation: Scheduling with the Agent for z/OS* documentation.

Figure 6. Distributed-driven workload environment for z/OS®



Dockerized environment

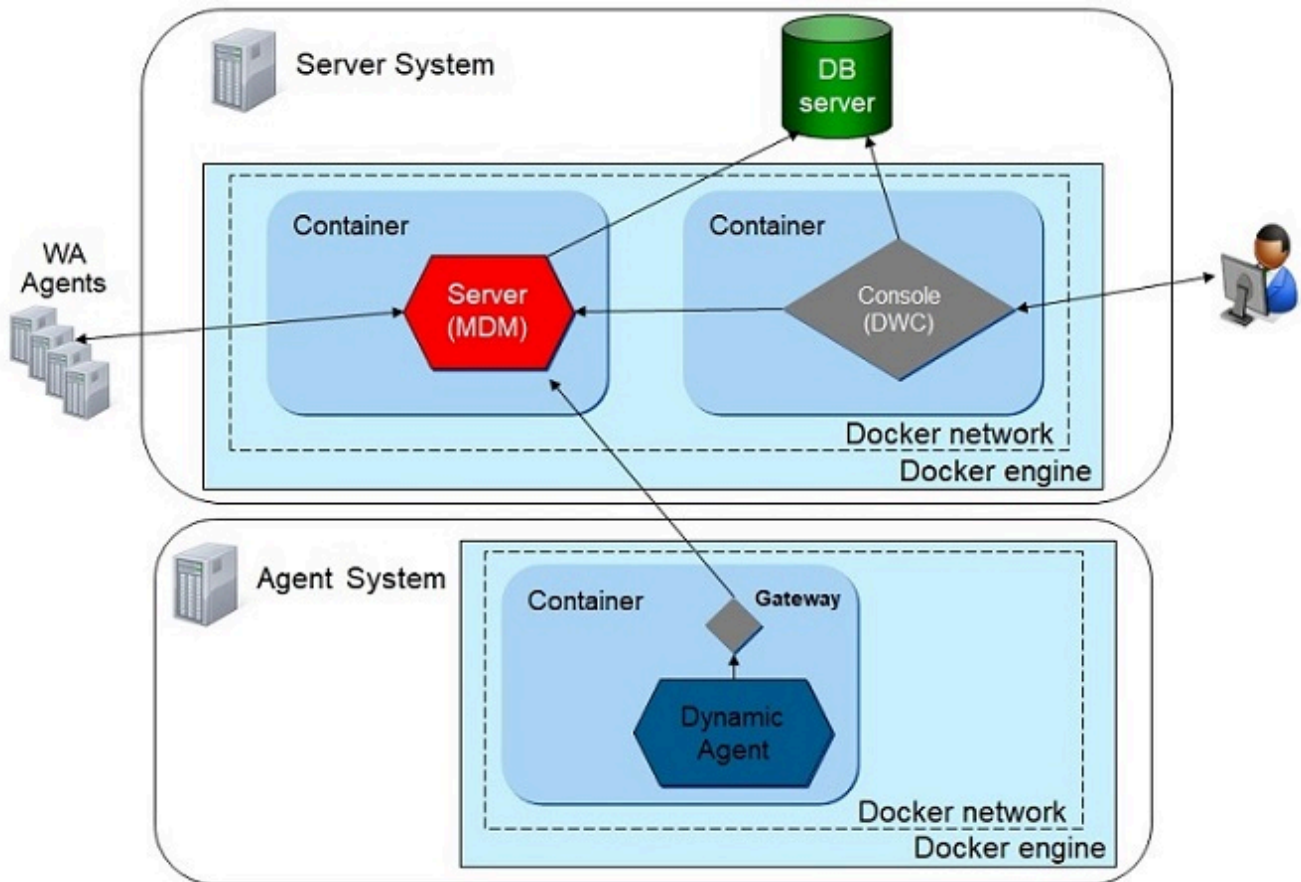
Use this configuration to implement a Dockerized environment.

Use this configuration to benefit of the HCL Workload Automation on a dockerized environment. Three containers are delivered and they can be deployed on the same engine or on different ones.

In the *Figure 1*, server and console components have been deployed on the same Docker engine and the dynamic agent component on a separated engine.

The database is always external to the Docker engine and a connection is established with server and console.

Figure 7. Dockerized environment configuration



Chapter 3. Planning domains

A HCL Workload Automation network contains at least one master domain manager that acts as a management hub for the product. Additional domains can be used to divide a widely-distributed network into locally-managed groups of workstations.

In a single domain configuration, the master domain manager maintains communications with all of the workstations in the network.

In a multiple domain configuration, the master domain manager communicates with the workstations in its domain and all immediately subordinate domain managers. The subordinate domain managers communicate with the workstations in their domains and their immediately subordinate domain managers, and so on. Domain managers report all of the activities of the domain to the master. Using multiple domains reduces network traffic and the load on the master by reducing the number of direct communications between the master domain manager and workstations. Multiple domains also provide fault-tolerance by limiting the outage caused by losing a domain manager in a single domain. To limit the effects further, you can designate backup domain managers to take over if domain managers fail.

When you define a new domain, you must identify the parent domain and the domain manager. The parent domain is the domain directly above the new domain in the domain hierarchy. All communications to and from a domain are routed through the parent domain manager.

Localized processing in your domain

Localized processing is separating your scheduling needs based on a common set of characteristics, such as geographical locations, business functions, and application groupings.

Group related processing can limit the amount of interdependency information that needs to be communicated between domains. The benefits of localized domains are:

Decreased network traffic

Keeping processing localized to domains eliminates the need for frequent inter-domain communication.

Tighter security and simplified administration

Security and administration can be defined at and limited to the domain level. Instead of network-wide or workstation-specific administration, you can have domain administration.

Optimized network and workstation fault-tolerance

In a multiple domain network, you can define backups for each domain manager so that problems in one domain do not disrupt operations in other domains.

Considerations in planning domains

There are a number of considerations that are to be taken into account when planning domains.

In planning your HCL Workload Automation network, consider the following:

Number of workstations, applications, and jobs

Consider the number of workstations that comprise the network and the number of applications and jobs that the network runs. If you have a small number of workstations, or a small number of applications to control, you do not need multiple domains.

Number of geographic locations

Consider the number of geographic locations covered by your network and the reliability and efficiency of communication between the locations. Multiple geographic locations is one of the primary reasons for choosing a multiple domain architecture. One domain for each geographical location is a common configuration. A single domain architecture relies on the network maintaining continuous processing.

Time zones

When your network is spread across multiple geographic locations in different time zones, decide whether to activate the time zone feature. See [Time zone considerations on page 28](#).

Centralized or decentralized management

You can manage single or multiple domain networks from a single master domain manager. If you want to manage multiple locations separately, you can consider the installation of a separate HCL Workload Automation network at each location. Some decentralized management is possible in a stand-alone HCL Workload Automation network by mounting or sharing file systems.

Types of applications

Consider the types of applications that are run by HCL Workload Automation. If you have multiple applications that are distinctly separate from each other, you might choose to put them in separate domains.

Windows™ network

When you have a Windows™ network, you might want your HCL Workload Automation domains to mirror your Windows™ domains.

System performance and other criteria

You can define multiple domains to localize systems based on performance or operating system type.

Amount of network traffic

If your network traffic is manageable, having multiple domains is less important.

Dependencies between jobs

Consider if you need to plan for job dependencies that cross system boundaries, geographical boundaries, or application boundaries. For example, does the start of Job1 on workstation1 depend on the completion of Job2 running on workstation2. The degree of interdependence between jobs is an important consideration when planning your network. If you use multiple domains, try to keep interdependent objects in the same domain to decrease network traffic and improve the use of the domain architecture. See *User's Guide and Reference*.

Level of fault-tolerance required

A disadvantage of the single domain configuration is the reliance on a single domain manager. In a multi-domain network, the loss of a single domain manager affects only the agents in its domain.

Firewalls

When your network contains firewalls, plan the structure of your domains around the firewalls. See *Administration Guide*.

Workstation classes

Workstations are organized into domains to make your network management easier and more efficient. However, the domain name is not one of the selection criteria when choosing where to run a job or job stream.

If you want to group workstations together because they have similar job scheduling characteristics, use a workstation class. Any number of workstations can be grouped in a class, and a workstation can be in many classes. Jobs and job streams can be assigned to run on a specific workstation class.

For example, you could set up workstation classes to group workstations according to:

- Your internal departmental structure, so that you could define a job that would be run on all the workstations in a department
- The software installed on them, so that you could define a job that would be run on all the workstations that had a particular application installed
- The role of the user, so that you could define a job that would be run on all the workstations belonging to, for example, managers

In this example, an individual workstation could be in one workstation class for its department, another for its user, and several others for the software installed on it.

Time zone considerations

Time zone support is an optional feature that is enabled by default.

It allows you to manage workloads at a global level. Time zone implementation also enables easy scheduling across multiple time zones.

For a description of how the time zone implementation works, see *User's Guide and Reference*.

For information about how to set the time zone implementation, see *HCL Workload Automation: Administration Guide*.

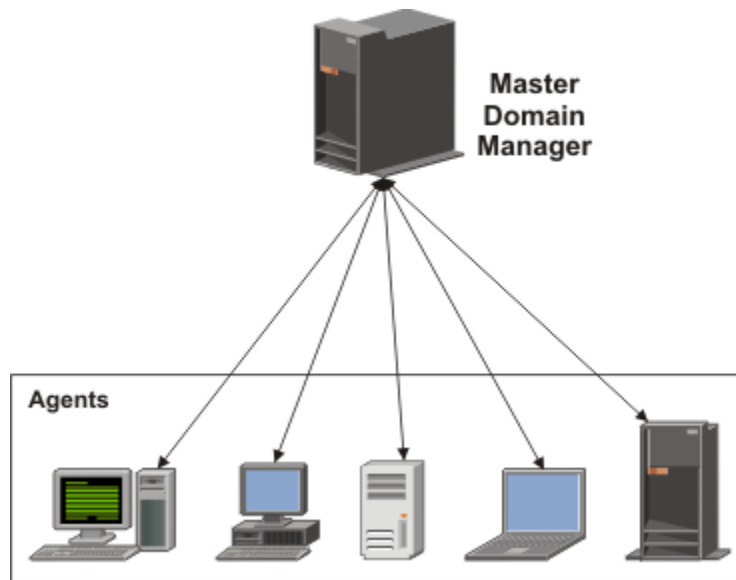
Single domain network

A single domain network consists of a master domain manager and any number of agents.

[Figure 8: Single domain topology on page 29](#) shows an example of a single domain network. A single domain network is well-suited to companies that have few locations and business functions. All communication in the network is routed

through the master domain manager. With a single location, you are concerned only with the reliability of your local network and the amount of traffic it can handle.

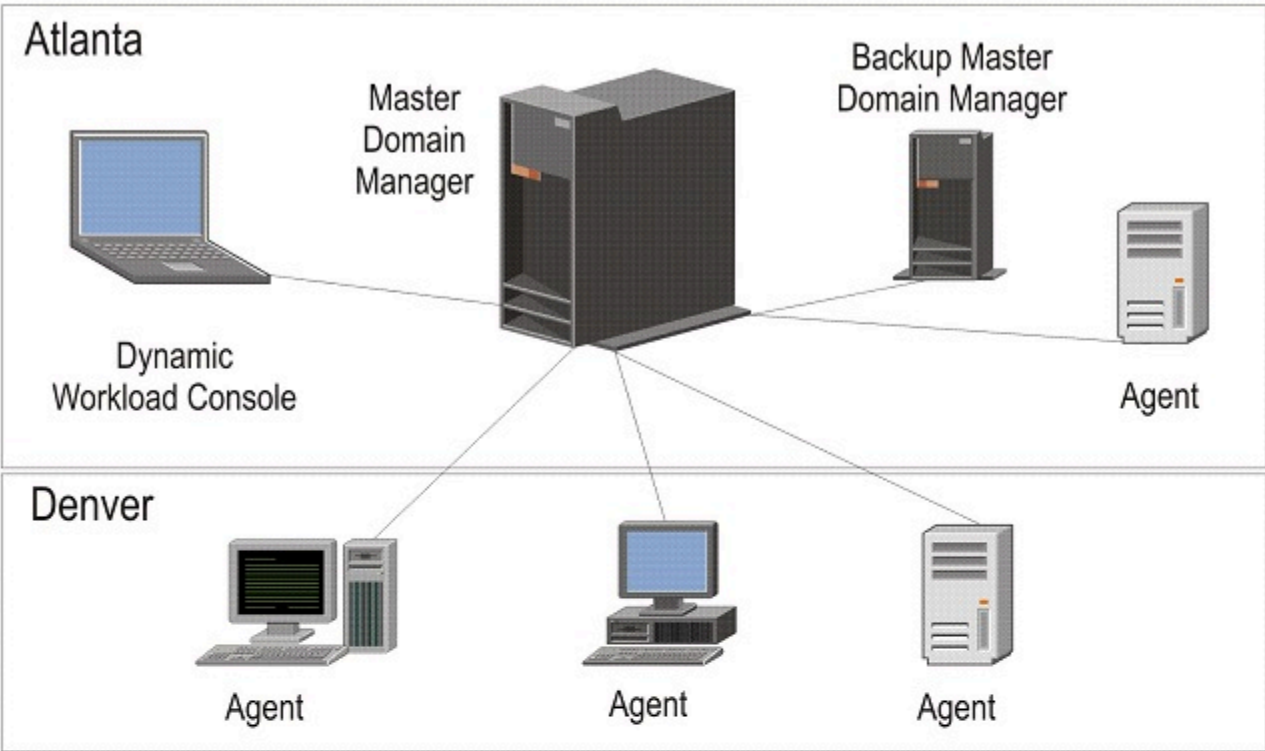
Figure 8. Single domain topology



Single domain networks can be combined with other networks, single or multiple domain, to meet multiple site requirements. HCL Workload Automation supports internetwork dependencies between jobs running on different networks.

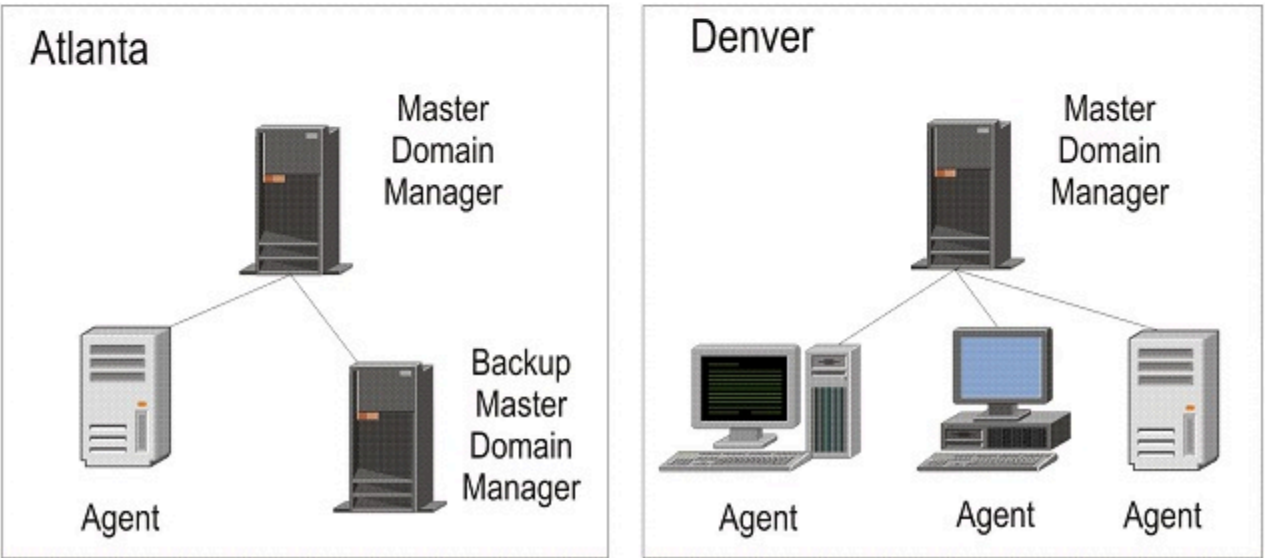
Figure 9. Single domain topology on multiple sites

Example 1



Or:

Example 2



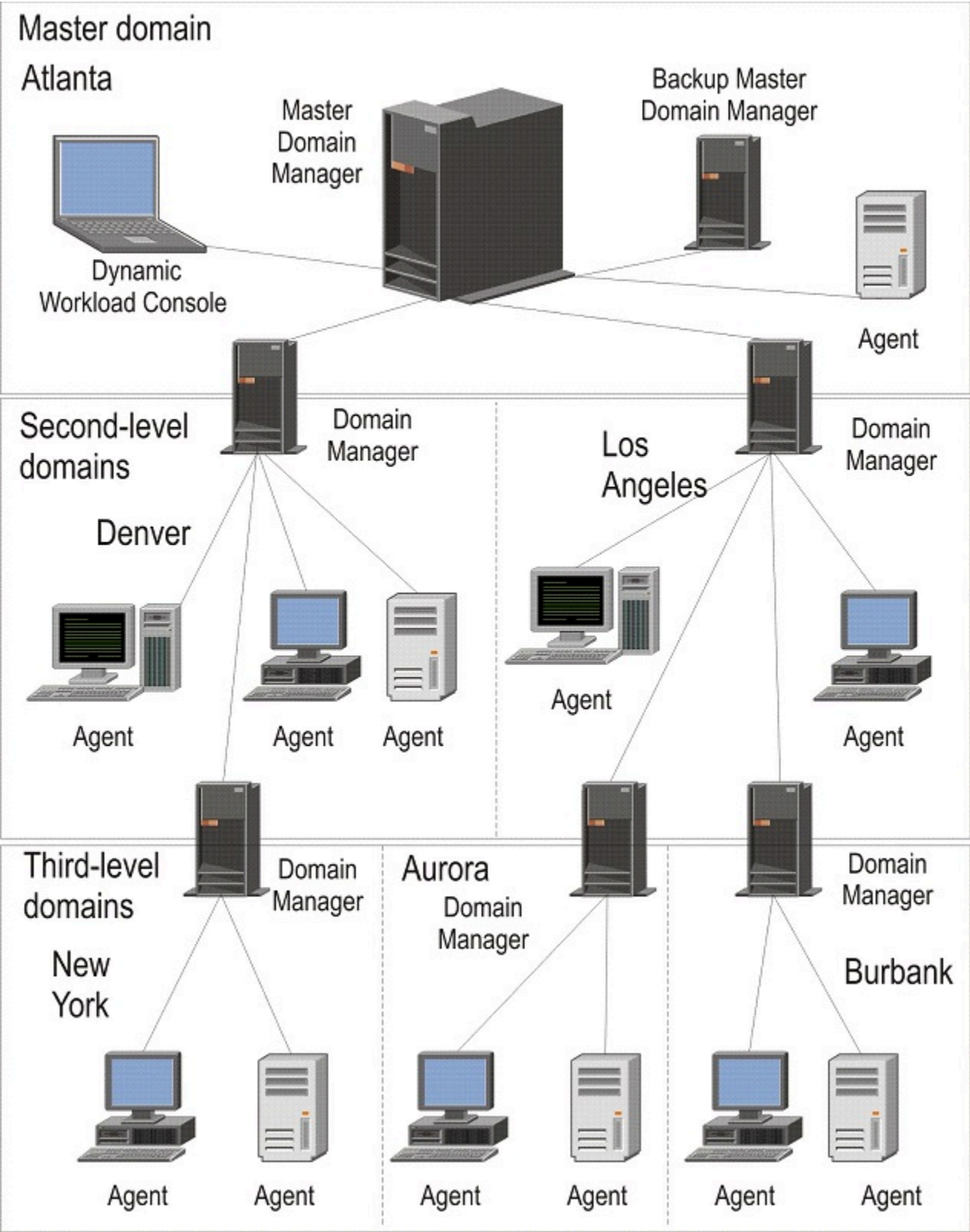
Example 1 shows a single domain network. The master domain manager is located in Atlanta, along with several agents. There are also agents located in Denver. The agents in Denver depend on the master domain manager in Atlanta to resolve all interagent dependencies, even though the dependencies might be on jobs that run in Denver. An alternative would be to create separate single domain networks in Atlanta and Denver, as shown in example 2.

Multiple domain network

Multiple domain networks are especially suited to companies that span multiple locations, departments, or business functions.

A multiple domain network consists of a master domain manager, any number of lower tier domain managers, and any number of agents in each domain. Agents communicate only with their domain managers, and domain managers communicate with their parent domain managers. The hierarchy of domains can go down to any number of levels.

Figure 10. Multiple domain topology



As [Figure 10: Multiple domain topology on page 32](#) illustrates, the master domain manager is located in Atlanta. The master domain manager contains the database files used to document the scheduling objects, and distributes the Symphony file to its agents and the domain managers in Denver and Los Angeles. The Denver and Los Angeles domain managers then distribute the Symphony file to their agents and subordinate domain managers in New York, Aurora, and Burbank. The master domain manager in Atlanta is responsible for broadcasting inter-domain information throughout the network.

All communication to and from the New York domain manager is routed through its parent domain manager in Denver. If there are schedules or jobs in the New York domain that are dependent on schedules or jobs in the Aurora domain, those dependencies are resolved by the Denver domain manager. Most inter-agent dependencies are handled locally by the lower tier domain managers, greatly reducing traffic on the network.

Chapter 4. Installation considerations

Some considerations that need to be taken into account before installation.

About this task

Before you begin the installation using the installation wizard, consider the following items that might apply to your specific environment.

Installing on Windows™ operating systems

If you are installing on Windows™, consider the following items.

- If you are using Windows™ Terminal Services, set the install user with the command: `change user /install`
- If TWS_USER is a domain user, Microsoft™ Computer Browser Service must be active.
- If TWS_USER is a domain user, the user performing the installation must be a domain administrator.

Remote installation

You cannot install HCL Workload Automation on a Windows™ workstation from a remote Samba-mounted file system.

Installing for end-to-end scheduling

If you are installing HCL Workload Automation on a workstation used as a distributed agent (that is either a standard agent, fault-tolerant agent, or domain manager) for end-to-end scheduling, specify OPCMASTER as the name of the master domain manager during the installation process. For further information about installing for end-to-end scheduling, see *Scheduling End-to-end with Fault Tolerance Capabilities*.

Create symbolic links

UNIX™ and Linux™. The installation wizard installs all executable files in its own `.bin` directory. Before running any HCL Workload Automation commands, you run a script that sets the command-line environment to access these files. To avoid having to set the environment each time you want to run any of the commands from within a script, you can select an installation option to create symbolic links to those commands or utilities most frequently used from within scripts. [Table 2: Symbolic link options on page 34](#) shows the binary paths and the symbolic links.

Table 2. Symbolic link options

TWS binary path	Symbolic link
<i>TWS_home/bin/at</i>	usr/bin/mat
<i>TWS_home/bin/batch</i>	usr/bin/mbatch
<i>TWS_home/bin/datecalc</i>	usr/bin/datecalc
<i>TWS_home/bin/jobstdl</i>	usr/bin/jobstdl

Table 2. Symbolic link options (continued)

TWS binary path	Symbolic link
<i>TWS_home/bin/maestro</i>	usr/bin/maestro
<i>TWS_home/bin/mdemon</i>	usr/bin/mdemon
<i>TWS_home/bin/morestdl</i>	usr/bin/morestdl
<i>TWS_home/bin/muser</i>	usr/bin/muser
<i>TWS_home/bin/parms</i>	usr/bin/parms

Installation paths

HCL Workload Automation is the name of a family of products and components, which includes the following:

- HCL Workload Automation
- HCL Workload Automation for Z
- HCL Workload Automation for Applications
- Dynamic Workload Console

Many HCL Workload Automation components are installed in what is called an *HCL Workload Automation instance*.

This section describes the installation paths of the HCL Workload Automation components:

***TWA_home* installation path**

Many of the components are installed in an HCL Workload Automation instance. Although this is a notional structure it is represented on the computer where you install HCL Workload Automation components by a common directory referred to in the documentation as *TWA_home*. The path of this directory is determined when you install an HCL Workload Automation component for the first time on a computer. You have the opportunity to choose the path when you make that first-time installation, but if you accept the default path, it is as follows:

On UNIX™ operating systems

```
/opt/wa/server_<wauser><n>
```

On Windows™ operating systems

```
%Program Files%\wa\server<n>
```

where *<n>* is an integer value ranging from 0 for the first instance installed, 1 for the second, and so on.

This path is called, in the publications, *TWA_home*. For details about the directories created outside of *TWA_home*, see the section about directories created outside of *TWA_home* in *Planning and Installation Guide*.

***TWA_DATA_DIR* and *DWC_DATA_dir* configuration directories**

To simplify administration, configuration, and backup and recovery on UNIX systems, a new default behavior has been implemented with regard to the storage of product data and data generated by HCL Workload Automation, such as logs and configuration information. These files are now stored by default in the `<data_dir>` directory, which you can optionally customize at installation time.

By default, this directory is *TWA_home*/TWSDATA for the server and agent components, and *DWC_home*/DWC_DATA for the Dynamic Workload Console. The product binaries are stored instead, in the installation directory.

You can optionally customize the `<data_dir>` directory at installation time by setting the `--data_dir` argument when you install using the command-line installation. If you want to maintain the previous behavior, you can set the `--data_dir` argument to the HCL Workload Automation installation directory.

If you deploy the product components using Docker containers, the `<data_dir>` is set to the default directory name and location, and it cannot be modified.

To retrieve the *TWA_DATA_DIR* and *DWC_DATA_dir* location in case you have modified the default path, check the values for the TWS_datadir and DWC_datadir properties stored in the `twainstance<instance_number>.TWA.properties` file. The file is located in `/etc/TWA`.

Alternatively, you can also proceed as follows:

1. Browse to `<TWA_home>/TWS` path.
2. Source the `./twc_env.sh` shell script.
3. Type `echo $UNISONWORK`. As a result, the path to the *TWA_DATA_DIR* is returned.

HCL Workload Automation installation directory

You can install more than one HCL Workload Automation component (master domain manager, backup master domain manager, domain manager, or backup domain manager) on a system, but each is installed in a separate instance of HCL Workload Automation, as described above.

The installation directory of HCL Workload Automation is:

```
<TWA_home>/TWS
```

***DWC_home* installation directory**

The Dynamic Workload Console can be installed in the path of your choice, but the default installation directory is as follows:

On Windows™ operating systems

```
%ProgramFiles%\wa\DWC
```

On UNIX™ operating systems

```
/opt/wa/DWC
```


On z/OS operating system

```
/opt/wa/DWC
```

HCL Workload Automation agent installation directory

The agent also uses the same default path structure, but has its own separate installation directory:

```
<TWA_home>/TWS/ITA/cpa
```



Note: The agent also installs some files outside this path. If you have to share, map, or copy the agent files (for example when configuring support for clustering) share, map, or copy these files, as well:

On UNIX™ operating systems

```
/etc/teb/teb_tws_cpa_agent_<twc_user>.ini
/opt/HCL/CAP/EMICPA_default.xml
/etc/init.d/tebctl-tws_cpa_agent_<twc_user>
(on Linux)
/etc/rc.d/init.d/tebctl-tws_cpa_agent_<twc_user>
(on AIX)
```

On Windows™ operating systems

```
%windir%\teb\teb_tws_cpa_agent_<twc_user>.ini
%ALLUSERSPROFILE%\HCL\CAP\EMICPA_default.xml
```

The agent uses the following configuration files which you might need to modify:

JobManager.ini

This file contains the parameters that tell the agent how to run jobs. You should only change the parameters if advised to do so in the HCL Workload Automation documentation or requested to do so by HCL Software Support. Its path is:

On UNIX™ operating systems

```
TWA_DATA_DIR/ITA/cpa/config/JobManager.ini
```

On Windows™ operating systems

```
TWA_home\TWS\ITA\cpa\config\JobManager.ini
```

JobManagerGW.ini

When a dynamic agent is installed and **-gateway** *local|remote* is specified, then this file contains the same parameters as the `JobManager.ini` file except for the following differences:

- The **ResourceAdvisorUrl** parameter points to the dynamic workload broker, and not the master domain manager.

The `JobManagerGW.ini` file is installed in the following location:

On UNIX™ operating systems

`TWA_DATA_DIR/ITA/cpa/config/JobManagerGW.ini`

On Windows™ operating systems

`TWA_home\TWS\ITA\cpa\config\JobManagerGW.ini`

ita.ini

This file contains parameters which determine how the agent behaves. Changing these parameters may compromise the agent functionality and require it to be reinstalled. You should only change the parameters if advised to do so in the HCL Workload Automation documentation or requested to do so by HCL Software Support. Its path is:

On UNIX™ operating systems

`TWA_DATA_DIR/ITA/cpa/ita/ita.ini`

On Windows™ operating systems

`TWA_home\TWS\ITA\cpa\config\ita.ini`

Installation path for files giving the dynamic scheduling capability

The files that give the dynamic scheduling capability are installed in the following path:

`<TWA_home>/TDWB`

The command line client installation path

The command line client is installed outside all *HCL Workload Automation instances*. Its default path is:

`TWA_home/TWS/CLI`

However, the information above supplies only the **default** paths. To determine the actual paths of products and components installed in HCL Workload Automation instances, see [Finding out what has been installed in which HCL Workload Automation instances on page 38](#)

Finding out what has been installed in which HCL Workload Automation instances

About this task

If you are not the installer of HCL Workload Automation and its components, you might not know what components have been installed, and in which instances of HCL Workload Automation. Follow this procedure to find out:

1. Access the following directory:

UNIX™ and Linux™ operating systems

`/etc/TWA`

Windows™ operating systems

`%windir%\TWA`

2. List the contents of the directory. Each HCL Workload Automation instance is represented by a file called:

`twainstance<instance_number>.TWA.properties`. These files are deleted when all the products or components in an instance are uninstalled, so the number of files present indicates the number of valid instances currently in use.

3. Open a file in a text viewer.



Attention: Do not edit the contents of this file, unless directed to do so by HCL Software Support. Doing so might invalidate your HCL Workload Automation environment.

The contents are similar to this on a master domain manager :

```
#TWAINstance registry
#Mon Feb 26 09:28:08 EST 2024
TWA_path=/opt/wa/server_twsuser
TWA_componentList=TWS
TWS_version=10.2.5
TWS_counter=1
TWS_instance_type=MDM
TWS_basePath=/opt/wa/server_twsuser/TWS
TWS_user_name=twsuser
TWS_wlpdir=/opt/wa/wlpEngine/wlp
TWS_datadir=/opt/wa/server_twsuser/TWSDATA
TWS_jdbcdir=/opt/wa/server_twsuser/TWS/jdbcdrivers/db2
```

The contents are similar to this on the Dynamic Workload Console:

```
#TWAINstance registry
Mon Feb 26 09:28:08 EST 2024
TWA_path=/opt/wa/DWC
TWA_componentList=DWC
DWC_version=10.2.5
DWC_counter=1
DWC_instance_type=DWC
DWC_basePath=/opt/wa/DWC
DWC_user_name=dwcadmin
DWC_wlpdir=/opt/wa/wlpDWC/wlp
DWC_datadir=/opt/wa/DWC/DWC_DATA
DWC_jdbcdir=/opt/wa/DWC/jdbcdrivers/db2
```

The important keys to interpret in this file are:

TWA_path

This is the base path, to which the installation added one or more of the following directories, depending on what was installed:

TWS

Where the HCL Workload Automation component is installed

DWC

Where the Dynamic Workload Console is installed

SSM

Where the Netcool® SSM monitoring agent is installed (used in event management)

TWA_componentList

Lists the components installed in the instance of HCL Workload Automation.

TWS_counter

Indicates if an HCL Workload Automation component is installed in this instance of HCL Workload Automation (when the value=1).

TWS_instance_type

Indicates which component of HCL Workload Automation is installed in this instance:

MDM

Master domain manager

BKM

Backup master domain manager

DDM

dynamic domain manager

FTA

Fault-tolerant agent or domain manager

TWS_user_name

The ID of the <TWS_user> of the HCL Workload Automation component.

TWS_wlmdir

The installation directory of the Open Liberty instance used by HCL Workload Automation.

TWS_datadir

The directory containing product data and data generated by HCL Workload Automation, such as logs and configuration information.

DWC_counter

Indicates if an instance of Dynamic Workload Console is installed in this instance of HCL Workload Automation (when the value=1)

DWC_user_name

The ID of the Dynamic Workload Console user.

DWC_wlmdir

The installation directory of the Open Liberty instance used by Dynamic Workload Console.

DWC_datadir

The directory containing product data and data generated by Dynamic Workload Console, such as logs and configuration information.

Directories created outside of *TWA_home* at installation time

The following list shows the directories that are created outside of *TWA_home* when you install HCL Workload Automation.

Windows operating systems

```
%WINDIR%\TWA

%WINDIR%\system32\TWSRegistry.dat (32 bits)
%WINDIR%\sysWOW64\TWSRegistry.dat (32 bits on 64 bits)
%WINDIR%\TWSRegistry.dat (64 bits on 64 bits)
%WINDIR%\teb
%WINDIR%\cit
%ProgramFiles%\tivoli\cit (or the path specified by %WINDIR%\cit\cit.ini)
```



Note:

During standalone dynamic agent installation, the `%WINDIR%\TWA` directory is not generated.

UNIX operating systems

A **root** installation of HCL Workload Automation results in the generation of the following directories outside of *TWA_home*:

```
/etc/TWA
/etc/TWS
/etc/teb
/etc/cit
/etc/init.d/tebclt-tws_cpa_agent_instance_name
/usr/Tivoli/TWS
/usr/ibm/tivoli/common/CIT/logs
/opt/tivoli/cit (or the path specified by /etc/tivoli/cit/cit.ini)
```

HCL Workload Automation also installs some files in the following existing folders:

```
/etc/systemd
/etc/rc.d
```



Note: If you want to check which files are stored in the **etc** directories, you can launch the following command: `find /etc -name "*<twuser>*"`

Contrary to **root** installations, **no-root** installations place generated directories within `<twuser_home>`, not within `/etc`.



Note:



During standalone dynamic agent installation, the `/etc/TWA` or `<twuser_home>/TWA` directory is not generated.

Windows™ services

When installing on the Windows™ operating system the Windows™ Service Control Manager registers services.

About this task

An installation on Windows™ operating systems registers the following services on the Windows™ Service Control Manager:

- HCL Workload Automation (for *TWS_user*)
- Netman (for *TWS_user*)
- Token Service (for *TWS_user*)
- HCL Workload Automation SSM Agent (for *TWS_user*)
- HCL Common Platform Agent: *twscpa_agent_* (for *TWS_user*)



Note: An existing service that has the same name as the new service will be overwritten during installation.

The Service Control Manager maintains its own user password database. If the *TWS_user* password is changed after installation, you must use the Services applet in the Control Panel to assign the new password for the Token Service and HCL Workload Automation (for *TWS_user*). For more information, see the section about changing the password of the *TWS_User* in *HCL Workload Automation: Administration Guide*.

Part II. Installing HCL Workload Automation

Available installation methods

About this task

This section provides the information required before you install the product. The available installation methods are listed, together with some considerations:

Advantages of the command-line installation

The command-line installation is a very simple procedure, which supports installing all components (master domain manager, backup domain manager, dynamic domain manager, backup dynamic domain manager, Dynamic Workload Console, and agents) using dedicated commands. You can choose to maintain the default values already defined in the properties file, specify all or part of the parameters in the command line when typing the command, or edit all or part of the parameters stored in the properties file. To proceed with the command-line installation, skip to [Installing from the command-line interface on page 45](#).

Advantages of the Docker deployment

The Docker installation is comprised of a set of pre-installed images for the master domain manager, the Dynamic Workload Console, and the DB2 database. All you have to do is launch the Docker installation commands.

Docker is a state-of-the-art technology which creates, deploys, and runs applications by using containers. Packages are provided containing an application with all of the components it requires, such as libraries, specific configurations, and other dependencies, and deploy it in no time on any other Linux or Windows workstation, regardless of any different settings between the source and the target workstation.

Docker adoption ensures standardization of your workload scheduling environment and provides an easy method to replicate environments quickly in development, build, test, and production environments, speeding up the time it takes to get from build to production significantly. Install your environment using Docker to improve scalability, portability, and efficiency.

To proceed with the Docker installation, skip to [Deploying containers with Docker on page 173](#).

Advantages of the Red Hat OpenShift deployment

The HCL Workload Automation product components can be deployed onto Red Hat OpenShift. You can deploy HCL Workload Automation components using certified containers on a Kubernetes-based container application platform useful to orchestrate containerized applications. You can then manage the HCL Workload Automation containers from the OpenShift dashboard or from the command line interface.

With OpenShift, you can implement distributed, advanced and scalable services based on the Docker container technology and orchestrated by Kubernetes. For more information, see [Deploying HCL Workload Automation components on Red Hat OpenShift using helm charts on page 176](#).

Advantages of deploying on Amazon EKS

To respond to the growing request to make automation opportunities more accessible, HCL Workload Automation is now offered on the Amazon Web Services cloud. Within just a few minutes, you can access the product Helm chart and container images and easily launch an instance to deploy an HCL Workload Automation server, console, and agents with full on-premises capabilities on AWS. HCL Workload Automation on AWS improves flexibility and scalability of your automation environment. It helps in lowering costs and eliminating complexity, while reducing the operational overhead and the burden involved in managing your own infrastructure, so you can invest your time and resources in growing your business. Also, HCL Workload Automation on AWS delivers faster access to managed services solutions, for a full product lifecycle management.

For more information see [Deploying on Amazon EKS on page 184](#).

Advantages of deploying on Azure Kubernetes Service (AKS)

You can use Azure AKS to deploy, scale up, scale down and manage containers in the cluster environment. Use the HCL Workload Automation Helm chart and container images to deploy the server, console and dynamic agent to the Azure AKS public cloud. Azure AKS gives you access to helpful services. For example, you can use the Azure SQL database, a highly scalable cloud database service. See [Deploying on Azure AKS on page 185](#) for more details.

Advantages of deploying on Google GKE

Google Kubernetes Engine (GKE) provides a managed environment for deploying, managing, and scaling your containerized applications using Google infrastructure. The Google GKE environment consists of multiple machines grouped together to form a cluster. You can also deploy and run Google Cloud SQL for SQL server.

Google GKE supports session affinity in a load balancing cluster, a feature which maintains each user session always active on the same pod. This ensures that the Dynamic Workload Console always connects to the same server during a session and that the user can perform any number of operations smoothly and seamlessly.

For more information, see [Deploying on Google GKE on page 185](#).

Advantages of deploying HCL Workload Automation to a cloud-native environment from the HCL Software Factory (SoFy).

HCL SoFy includes all the tools required to build a Kubernetes deployment package for HCL Workload Automation, and run it on the Kubernetes cloud of your choice (public or private).

HCL SoFy uses Helm technology to provide HCL products and application programming interfaces. A temporary sandbox environment is provided to deploy and test solutions. You can run more than two temporary sandboxes at the same time. See [Workload Automation on HCL SoFy on page 186](#) for more details.

Chapter 5. Installing from the command-line interface

About this task

Install HCL Workload Automation from the command-line interface based on a typical installation scenario. Variations to the typical scenario are described in the FAQ sections.

Before you get started, download the installation images and verify the prerequisites, as described in sections [Downloading installation images on page 45](#) and [Prerequisites on page 45](#).

Downloading installation images

Steps to take when downloading images on your workstation.

About this task

You can download installation images from [HCL Software](#).

1. Ensure that your workstation has sufficient space to store the compressed file containing the installation images. For more information about system requirements, see the product requirements in the online documentation..
2. From [HCL Software](#), download the compressed file, containing the latest product image, to a temporary directory.
3. Extract the installation image from the downloaded file and verify that the installation image is complete. Extract the content of the ZIP files into a directory, using one of the extraction tools available on your system or that can be downloaded from the internet. The tool you use must be able to keep the file permissions on the extracted files, for example, `infozip`.

On Windows™ systems, ensure that you extract the image into a path that is not very long, otherwise, the file name might be truncated. The maximum length allowed is 255 characters.

If you are installing on a UNIX™ operating system, run the following command:

```
chmod -R 755 <imagesDir>
```



Note: To extract the `.zip` file onto a Windows™ 64-bit system, ensure that the image is not located on the desktop because the Windows™ operating system extract tool might encounter a problem. Choose another directory into which to extract the product image.



Note: DB2 is available for download from [HCL License Portal](#) only. The latest versions of Open Liberty can be downloaded from [Get started with Open Liberty](#). For further details, see the `HWA_10.2.5_QuickStartGuide.zip` available from [HCL Software](#).

Prerequisites

When installing HCL Workload Automation components, consider the following prerequisites.

About this task

For a complete list of the correct versions to install, see the System Requirements Document at [HCL Workload Automation Detailed System Requirements](#).

For a complete list of system requirements (disk spaces, temporary spaces and RAM usage), see [HCL Workload Automation Detailed System Requirements](#).

Open Liberty

The latest versions of Open Liberty can be downloaded from [Get started with Open Liberty](#).

If you already have WebSphere Application Server Liberty Base installed, you can use it with HCL Workload Automation, otherwise you can install Open Liberty, as described in [Installing Open Liberty on page 56](#).

WebSphere Application Server Liberty Base is no longer distributed from the HCL Software portal and needs to be obtained independently.

If you want to move from WebSphere Application Server Liberty Base to Open Liberty, see the topic about moving from WebSphere Application Server Liberty Base to Open Liberty in *Administration Guide*.

Before you install HCL Workload Automation for the first time, you must have one of the following databases installed. The following requirements apply to the RDBMS systems:

DB2

DB2® Enterprise Server Edition

You can install DB2® Server and the master domain manager or Dynamic Workload Console on the same workstation, then configure the database drivers from any workstation in your environment.

If you purchase a new DB2 license, you can typically use your existing DB2 installation without needing to reinstall the software. You simply need to apply the new license certificate to your current DB2 setup using the db2licm command to update your license compliance. For more information, see [Applying Db2 licenses](#). If you have any further questions regarding your license on your account, contact your sales representative.

You can install DB2® manually.

Oracle and Amazon RDS for Oracle

You can install Oracle in the following ways:

Oracle Enterprise Edition

The advantage of choosing Oracle Enterprise Edition is that you can implement the Oracle Partitioning feature to improve the performance of event-driven workload automation. This improves rule management performance, in particular the following queries: event_rule_instance, action_run, and operator_messages. For information about event-driven workload automation, see the section about event-driven workload automation in *User's Guide and Reference*.

Oracle Standard Edition

Oracle Standard Edition does not include the Oracle Partitioning feature. Installing this edition does not improve the performance of event-driven workload automation.

Amazon RDS for Oracle

Amazon RDS for Oracle is a robust and convenient option for managing Oracle databases in the cloud. It handles routine database tasks such as provisioning, patching, backup, recovery, and scaling, allowing you to focus on application development.

For supported versions, see the HCL Workload Automation System Requirements Document at [HCL Workload Automation Detailed System Requirements](#).



Note:

- When installing the product on a 64-bit library operating system, use an Oracle database on a 64-bit library.
- When upgrading:
 - If you already have an RDBMS installed and you want to upgrade it, you must upgrade it **after** you upgrade HCL Workload Automation.
 - Use an Oracle database on a 64-bit library when installing the product on a 64-bit library.

For information about upgrading the RDBMS, see the data maintenance chapter in the *Administration Guide*.

MSSQL

Before you create the HCL Workload Automation schema on the database, you must have created the directory where the HCL Workload Automation table spaces will be placed when the HCL Workload Automation schema is created. The default is `C:\MSSQL`.

Azure SQL

A family of managed, secure, and intelligent products that use the SQL Server database engine in the Azure cloud

Google Cloud SQL for SQL server

A fully-managed database service that helps you set up, maintain, manage, and administer your relational databases on Google Cloud Platform.

Amazon RDS for MSSQL

Amazon RDS for MSSQL offers a powerful and user-friendly solution for managing MSSQL databases in the cloud. It takes care of routine tasks like provisioning, patching, backups, recovery, and scaling, so you can concentrate on developing your applications.

PostgreSQL

A powerful, open source object-relational database system, which provides reliability, feature robustness, and performance.

Local user

The installation of HCL Workload Automation requires the creation of a local user. For more information, see [Creating the HCL Workload Automation administrative user on page 99](#).

Scanning system prerequisites for HCL Workload Automation

Before installing or upgrading the product, HCL Workload Automation automatically runs a scan on your system.

Before you begin

When installing HCL Workload Automation using the `serverinst` script, the script first runs the scanner to verify system prerequisites.



Note: To ensure that the prerequisite scan process does not fail, verify that the `bc` executable is present on the local system and that it is set in the PATH environment variable. If you do not want to install the `bc` executable, you can skip the prerequisites check by using the `skipcheckprereq` parameter when running the `serverinst` and `twinst` parameters. For more information about the `bc` executable, see [bc, an arbitrary precision calculator language](#). For more information about installation commands, see [Server components installation - serverinst script on page 442](#) and [Agent installation parameters - twinst script on page 119](#).

About this task

Having an environment that meets the product system requirements ensures that an installation or upgrade succeeds without any delays or complications.

The scan verifies that:

- The operating system is supported for the product.
- On UNIX™ operating systems, the necessary product libraries are installed.
- There is enough permanent and temporary disk space to install both the product and its prerequisites.
- There is enough memory and virtual memory.



Note: The scan verifies only that the environment meets the requirements of HCL Workload Automation. It does not check the requirements for other components, such as DB2®.

If any of these checks fails, HCL Workload Automation returns an error message.

The log files for the server components are located in:

On Windows™ operating systems:

```
<TWA_home>\logs\serverinst<version_number>.log
```

On UNIX™ and Linux™ operating systems:

```
<TWA_DATA_DIR>/installation/logs/serverinst<version_number>.log
```

The log files for the Dynamic Workload Console are located in:

On Windows™ operating systems:

```
<DWC_home>\logs\dwcinst<version_number>.log
```

On UNIX™ and Linux™ operating systems:

```
<DWC_DATA_dir>/installation/logs/dwcinst<version_number>.log
```

The log files for the agents are located in:

On Windows™ operating systems:

```
<TWA_home>\logs\twsinst<interp><user_name><version_number>.log
```

On UNIX™ and Linux™ operating systems:

```
<TWA_DATA_DIR>/installation/logs/twsinst<interp><user_name><version_number>.log
```

You can decide to rerun the installation or upgrade without executing the prerequisite scan. If you set the **-skipcheckprereq** parameter to `true` when performing the installation, the installation script does not execute the prerequisite scan. If a problem occurs, an error is displayed, the component is installed or upgraded, but might not work. For more information about the `-skipcheckprereq` parameter in all installation scripts, see the reference section in the *HCL Workload Automation: Planning and Installation*.

HCL Workload Automation user management

The HCL Workload Automation user management on UNIX and Windows operating systems

About this task

Consider the following constraints and properties for the HCL Workload Automation user:

On Windows operating systems:

The installation process automatically creates the HCL Workload Automation user. If your security policies do not allow user creation during the installation process, create the user and give it the necessary right as described in [Windows user domain rights and structure on page 50](#).

On UNIX and Linux operating systems:

Regardless of the method of installation you choose, the HCL Workload Automation user must be created manually before running the installation and must be enabled to login to the machine where the master domain manager is going to be installed. Use the appropriate UNIX™ and Linux™ operating system commands to create the user.

You can choose to install with the **root user** or you can perform a **no-root installation**, using a user without root privileges. Note that if you perform a no-root installation, only the user who performs the installation can use HCL Workload Automation. When uninstalling, use the same user who performed the installation.



Note: Some operating systems require that for users with a password, the password must be changed at the first login. If this is the case, for a successful installation, you will need to log in as the user and change the password for the first time.

Windows™ user domain rights and structure

About this task

If you install on Windows™ operating systems, consider the following information.

For the installation:

- You cannot have a local user and a domain user with the same name. For example, you cannot have **user1** as local user and at the same time **user1@domain1** and **domain\user1**.
- The Windows™ user performing an agent installation and the user that will own the instance of HCL Workload Automation must:
 - For a local HCL Workload Automation user, be a member of the local administrative group
 - For a domain HCL Workload Automation user, be a member of the domain "users" group in the domain controller and be a member of the local administrative group.

For Windows™ HCL Workload Automation users:

All Windows™ HCL Workload Automation users must have the following user permissions. They can be granted locally. Domain level policies always override local policies, so you might be required to grant the permissions from the domain:

- Act as part of the operating system
- Allow log on locally
- Impersonate a client after authentication
- Log on as a batch job
- Log on as a service
- Replace a process level token
- Adjust memory quotas for a process (available on some configurations only)



Note: These rights are granted during the installation, but you can confirm them manually.

To run HCL Workload Automation command lines:

On Windows operating systems with UAC disabled:

In addition to standard Windows permissions, to log on to the machine, the user must have the "Impersonate a client after authentication" permission granted. By default, this is granted just to

the "Administrators" group members. This permission is required to impersonate the TWS user and access the HCL Workload Automation Mailbox.

On Windows operating systems with UAC enabled:

This is the default value. The "Impersonate a client after authentication" is not available to the user, unless the cmd shell is started with "Run as administrator" permission. To run HCL Workload Automation command lines, the user must have "Impersonate a client after authentication" permission defined and then start the shell with the "Run as administrator" permission authenticating with its own user ID.

For the Streamlogon user:

The user must have the "logon as batch" permission to allow HCL Workload Automation to create the job process. In addition, you must assign to the user "Read" and "Read & execute" permission to cmd.exe. You can assign "Read" and "Read & execute" permission to cmd.exe also to the BATCH built-in group instead of to a single user.

To manage HCL Workload Automation agents:

The user must be in the Administrators group or must be able to perform "Run as" as **twuser** to reset the HCL Workload Automation files if a recovery is needed.

Considerations for Windows™ domain controllers running Microsoft™ Active Directory

If you want to install a HCL Workload Automation fault-tolerant agent on workstations where users who run jobs are domain users and the domain controller is running the Microsoft™ Active Directory, decide how to install the agents and configure the domain to have the jobmon process obtain the correct information to allow the users to run jobs.

About this task

Before running a job, jobmon retrieves information about the user running the job. If the user is a domain user and the domain controller is running Microsoft™ Active Directory, whether the user information can be retrieved depends on the information in the access control list (ACL) of that user. The main jobmon process that runs the job is started as the local system account (AUTHORITY\SYSTEM), but it immediately impersonates the *TWS_user* that owns the fault-tolerant agent. This means that for jobmon to successfully launch the job, the *TWS_user* must have an access control entry (ACE) in the ACL of the user for which it is trying to retrieve information.

Perform one of the following actions:

Enable the *TWS_user* to access a set of users that run jobs

On the domain server, edit the ACL of all users that run jobs on the workstation and add an ACE for each *TWS_user*. In this case, only specified users can run the jobs submitted by jobmon.

Allow all users to run jobs submitted by jobmon by using the *TWS_BYPASS_DC=TRUE* system variable

Create the *TWS_BYPASS_DC=TRUE* system variable, with a value not null, and reboot the workstation. In this case, jobmon obtains the user information without performing the security check for the ACE in the ACL of the user. All the local and domain users can run the jobs submitted by jobmon.

Allow all users to run jobs submitted by jobmon by setting the *TWS_user* as a domain user

Set up the *TWS_user* as a Windows™ domain user and install the instance of HCL Workload Automation using the *TWS_user*. In this case, all authenticated users on the domain controller can access the default ACL for a domain user. Jobs can then be launched by both local and the domain users. All the local and the domain users can run the jobs submitted by jobmon.

Exclude the workstation from the security check on users ACL

On the domain server, add the host name of the workstation where the fault-tolerant agent is installed to the Pre-Windows 2000-Compatible Access Group. In this way, from a security point of view, the domain controller interacts with this workstation as if it is in a Windows™ domain that does not support Active Directory. In this case, all the local and domain users can run the jobs submitted by jobmon. In addition, the domain controller does not prevent any local or domain users from running other processes that are not controlled by HCL Workload Automation.

Enabling product license management

Before you can use the product, you need to define a license server, configure the master domain manager to contact it, and map the license entitlements at [My HCLSoftware](#).

HCL Workload Automation uses a pay-per-use licensing model. Statistics of successfully completed jobs are stored on the licensing server for monitoring and compliance verification.

Users of the software will be authorized to use the software up to the allocation of entitlement you made on the server.

Before you get started, you should become familiar with the following concepts:

Permanent license

A license which expires when you have consumed all the jobs you are entitled to, with no expiration date.

Term license

A license which expires on a set date, also if you have not consumed all the jobs you are entitled to.

Additional resources

- For a general overview about My HCLSoftware, see [My HCLSoftware - an overview](#).
- For detailed information about My HCLSoftware, see [What is My HCLSoftware?](#) and [How to register as a Customer on HCLSoftware portals](#).

License mapping

You can assign to a device a quota or the total quantity of licenses available from one or more entitlements. Each HCL Workload Automation master domain manager can contact only one device, while multiple master domain managers can contact the same devices.

All master domain managers contacting the same device are contributing to consume the assigned quantity. A strategy to limit consumption for a specific HCL Workload Automation environment is to create a separate device for the environment and assign to it only the desired quota of the available licenses.

If you have a permanent license, ensure you map also a term license in the main environment.



Note: Assigning a license to a device is a permanent operation and cannot be undone.

High-level procedure

After installing HCL Workload Automation, perform the following steps:

1. Log in to [My HCLSoftware](#) using the credentials provided upon purchasing HCL Workload Automation.
2. Browse to the **Deployments** page.
3. Click **Add Deployment** to create a new deployment.
4. Specify a name for your deployment and select HCL Workload Automation in the pull-down menu.
5. Click the three dots in the tile for your deployment.
6. Select **Create Deployment Key**.
7. Save the key before closing the dialog box. This key is unique to your deployment and cannot be retrieved after closing the dialog box. If you lose the key, you will have to create a new one, at which point this one will be invalidated. Specify this key when setting up the **licenseRefreshToken** option.
8. Connect HCL Workload Automation to the license server by configuring the relevant global options using the `optman` command, for example:
 - to configure the URL address (**licenseServerUrl**), run:

```
optman chg lu=URL_address
```

- to configure the refresh token (**licenseRefreshToken**), run:

```
optman chg rt=token_value
```

The full list of relevant global options is as follows:

- **licenseProxyServer**
- **licenseProxyServerPort**
- **licenseProxyUser**
- **licenseProxyPassword**
- **licenseRefreshToken**
- **licenseServerUrl** The URL value is <https://api.hcltechsw.com/>.

For more information about setting global options, see [Global options - detailed description](#) the section about setting global options in *Administration Guide*

9. If the master domain manager cannot connect to the Internet, you can set up a proxy server to allow the master domain manager to contact the license server using the **licenseProxyPassword** and **licenseProxyServer** global options.
10. After completing the configuration, you can check your licenses in the **Subscriptions** tab.

License server IDs are unique and cannot be shared.

You can change the number of license servers and the related license assignments at any time by accessing [My HCLSoftware](#).

For more information about how licenses are tracked and how to generate a report that summarizes your monthly per-job license usage, see the section about license computation model in *Administration Guide*.

Troubleshooting

To verify the connection to the MHS server, run the following command:

```
curl -I https://api.hcltechsw.com
```

If you receive an error message stating that your job count is not being sent to the server due to a problem with the refresh token, generate a new refresh token and specify it again in the **licenseRefreshToken** option, as described in [High-level procedure on page 53](#).

Typical installation scenario

Scenario for a fresh typical installation at the latest product version of HCL Workload Automation

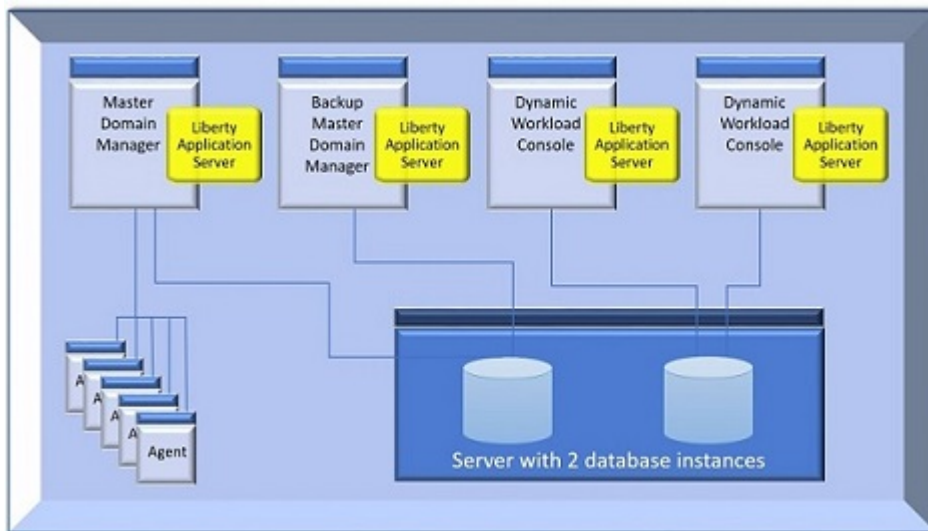


This scenario describes how to perform a fresh install at the latest product version of the full software stack for HCL Workload Automation, which consists of the following components and workstations:

- One workstation for the database server which hosts both the master domain manager and Dynamic Workload Console databases.
- One workstation for the master domain manager and the related Open Liberty.
- One workstation for the backup master domain manager and the related Open Liberty. The master domain manager and backup master domain manager share the same database. This ensures the backup master domain manager has the latest data and can take over seamlessly, in case the master domain manager fails.
- Two workstations for two Dynamic Workload Console installations, each of them with their related Open Liberty. The two Dynamic Workload Console instances share the same database.
- A number of agents.

[Figure 11: Typical HCL Workload Automation architecture on page 55](#) describes how the HCL Workload Automation components listed above are usually installed.

Figure 11. Typical HCL Workload Automation architecture



If you already have WebSphere Application Server Liberty Base installed, you can use it with HCL Workload Automation, otherwise you can install Open Liberty, as described below.

Starting from version 10.2.1, using certificates is mandatory when installing or upgrading the product. You can use the `certman` command to extract certificates from an existing keystore, generate new certificates from a new or existing CA, and much more. For more information, see the topic about managing certificates using `certman` in *Administration Guide*.

If you install the master domain manager on recent UNIX operating systems, you can use the OpenSSL 3.0.x libraries provided with the operating system. The list of UNIX operating systems whose libraries you can use is as follows:

- Ubuntu 22
- AIX 7.3
- Red Hat 9

To ensure HCL Workload Automation uses these libraries, always launch the installation or upgrade procedure from a brand new shell. You can also check the OpenSSL library currently in use with the `which openssl` command and check the OpenSSL version with the `openssl version` command.

This release installs a new version of the file `twc_env.sh` (`twc_env.cmd`) and also creates a backup file named, `twc_env.sh.bk` (`twc_env.cmd.bk`), which are both saved to the `TWA_HOME/TWC` directory, where `TWA_HOME` is the HCL Workload Automation installation directory. After completing the installation, if you have modified the original version, merge the content of the new version with the content of the customized version to carry your customized content into the new version. When merging the two versions as described above, ensure you do not modify the paths to OpenSSL libraries.

You can now proceed to [Installing Open Liberty on page 56](#).

Installing Open Liberty

Open Liberty is required on all workstations where you plan to install the server components and the Dynamic Workload



Before you begin

On AIX and Linux workstations, ensure you permanently set the **ulimit** parameter as follows:

- data segment process (option **-d**) = unlimited
- file size (option **-f**) = unlimited
- max user processes (option **-u**) = >260000 up to unlimited
- open files (option **-n**) = >100000 up to unlimited
- max memory size (option **-m**) = unlimited
- stack size (option **-s**) = >33000 up to unlimited

On the master domain manager, these settings must be applied to:

- root
- the HCL Workload Automation administrative user

On the Dynamic Workload Console, these settings must be applied to:

- root
- the Dynamic Workload Console installation user (if this user is different from root)

Ensure that your system meets the operating system requirements. For more information, see Open Liberty detailed system requirements.

About this task

You can quickly install Open Liberty by extracting an archive file on all supported platforms.

If you already have WebSphere Application Server Liberty Base installed, you can use it with HCL Workload Automation, otherwise you can install Open Liberty, as described below.

If you want to move from WebSphere Application Server Liberty Base to Open Liberty, see the topic about moving from WebSphere Application Server Liberty Base to Open Liberty in *Administration Guide*.

Install Open Liberty on all of the following workstations, which comprise a typical installation:

- master domain manager
- backup domain manager
- two Dynamic Workload Console installations on two separate workstations

If you plan to install a dynamic domain manager and its backup, these components require a separate Open Liberty installation.

On UNIX workstations, you can install Open Liberty using a user of your choice. In this case, assign the HCL Workload Automation administrative user read and write access to the Open Liberty installation directory.

To install Open Liberty, perform the following steps:

1. Find out which version of Open Liberty is required, by checking the required version of the Application server in the **Supported Software Report**, available in Product Requirements.
2. Download Open Liberty from [Get started with Open Liberty](#). Download the package named **All GA Features**
3. Perform one of the following actions:
 - a. Extract Open Liberty using the root user:

On Windows operating systems

```
unzip <openliberty_download_dir>\openliberty-<version>.zip
-d <install_dir>
```

On UNIX operating systems

```
unzip <openliberty_download_dir>/openliberty-<version>.zip
-d <install_dir>
```

- b. Run the following command to assign permissions:

```
chmod 755 -R "wlp_directory"
```

OR

Extract Open Liberty using the user who is going to install the product, as follows:

```
su - "wauser"
unzip
```

where:

<openliberty_download_dir>

The directory where you downloaded Open Liberty.

install_dir

The directory where you want to install Open Liberty.



Note: Install the new Open Liberty in the exact location of the previous WebSphere Application Server Liberty Base installation.

4. Ensure the HCL Workload Automation administrative user has the rights to run Open Liberty and full access to the installation directory. If Open Liberty is shared between the master domain manager and the Dynamic Workload Console, ensure also the Dynamic Workload Console user has the same rights.

Results

You have now successfully installed Open Liberty.

What to do next

You can now proceed to [Encrypting passwords \(optional\)](#) on page 58.

Encrypting passwords (optional)

How to encrypt passwords required by the installation, upgrade, and management processes.

About this task



-passphrase

Specifies the custom passphrase that is used to generate the key with which the command encrypts the password. If you set this parameter, inform the user who installs HCL Workload Automation that they must define the **SECUREWRAP_PASSPHRASE** environment variable in the same shell from which they run the installation command, and set it to the same value as the **passphrase** parameter. On Windows operating systems, the passphrase must be at least 8 characters long. This argument generates a password which can be reused for all HCL Workload Automation components. This parameter is mutually exclusive with the [-useaeskeystore on page 429](#) parameter, which generates a password which can be decrypted only on the local workstation and not reused for other components.

3. Provide both the encrypted password and custom passphrase to the user in charge of installing HCL Workload Automation. You can use encrypted passwords only in association with the specific passphrase used to encrypt them.

Installing with the encrypted password

The user in charge of installing HCL Workload Automation must set the **SECUREWRAP_PASSPHRASE** environment variable by performing the following steps:

1. Open a brand new shell session.
2. Ensure that no value is set for the **SECUREWRAP_PASSPHRASE** environment variable.
3. Define the **SECUREWRAP_PASSPHRASE** environment variable and set it to the passphrase defined by the user who ran the secure command, as follows:

```
SECUREWRAP_PASSPHRASE=<passphrase>
```

You can use encrypted passwords only in association with the specific passphrase used to encrypt them.

4. In the same shell session, provide the encrypted passwords when running any command that uses a password. An encrypted password looks like the following example:

```
{aes}AFC3jj9cR0YyqR+3CONBzVi8deLb2Bossb9GGroh8UmDPGikIkzXZzid3nzY0IhnSg=
```

What to do next

You can now proceed to [Creating and populating the database on page 59](#).

Creating and populating the database

Create the required databases before you begin the installation.



Before you start the installation, you must create and populate the database for both the master domain manager and the Dynamic Workload Console. You can perform a typical database procedure, as described in the following scenarios,

or you can customize the database parameters, as described in [FAQ - Database customizations on page 87](#). Links to customization options which are specific for a single database, if any, are provided in the related scenario.

The procedure differs for each supported database, as listed below:

Db2

- [Creating and populating the database for DB2 for the master domain manager on page 61](#).
- [Creating and populating the database for DB2 for the Dynamic Workload Console on page 63](#)

Db2 for z/OS

- [Creating and populating the database for DB2 for z/OS for the Dynamic Workload Console on page 66](#)



Note: If you use Db2 for z/OS with the Dynamic Workload Console version 10.2.4 or later, transfer the drivers in binary mode from the directory where you installed Db2 for z/OS to a directory of your choice. When you run the `configuredb` or `dwcinst` script, set the directory you chose in the **`dbdriverspath`** parameter.

Oracle

- [Creating the database for Oracle and Amazon RDS for Oracle for the master domain manager on page 69](#)
- [Creating the database for Oracle and Amazon RDS for Oracle for the Dynamic Workload Console on page 72](#)

MSSQL

- [Creating the database for MSSQL for the master domain manager on page 74](#)
- [Creating and populating the database for MSSQL for the Dynamic Workload Console on page 76](#)

MSSQL cloud-based databases

- [Creating the database for MSSQL cloud-based databases for the master domain manager on page 79](#)
- [Creating and populating the database for MSSQL cloud-based databases for the Dynamic Workload Console on page 81](#)

PostgreSQL

- [Creating and populating the database for PostgreSQL or PostgreSQL cloud-based databases for the master domain manager on page 82](#)
- [Creating and populating the database for PostgreSQL or PostgreSQL cloud-based databases for the Dynamic Workload Console on page 85](#)

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.

On UNIX™ operating systems, ensure the database administrator has read and write privileges for the HCL Workload Automation installation path.



Note:

1. If you create the schema on your own, ensure the COLLATE value is set appropriately. Consider the following examples:

Db2

```
db2 get db cfg for TWS | grep -i collating
```

The expected values are:

```
Database collating sequence = IDENTITY
Alternate collating sequence (ALT_COLLATE) =
```

MSSQL

```
select DATABASEPROPERTYEX('Your DB Name','collation')
```

The expected values is:

```
Latin1_General_BIN2
```

2. During database upgrade, the **CUR_COMMIT** configuration parameter is set to **DISABLED** to maintain the same behavior as in previous releases. For the proper functioning of HCL Workload Automation and to prevent possible internal deadlocks, set the **CUR_COMMIT** parameter to **ON**.

Creating and populating the database for DB2 for the master domain manager

Instructions for creating and populating the HCL Workload Automation database for DB2 for the master domain manager

About this task



You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in [FAQ - Database customizations on page 87](#).

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

DB2 requires a specific procedure in which you first create the database and then create and populate the database tables. To simplify the database creation, a customized SQL file named `create_database.sql` is provided, containing the

specifics for creating the HCL Workload Automation database. The database administrator can use this file to create the database. After the database has been created, you can proceed to create and populate the database tables.

You can optionally configure DB2 in SSL mode on UNIX operating systems by specifying the **sslkeysfolder** and **sslpassword** parameters when you run the `configureDb` command. For more information, see the topic about using certificates when DB2 or PostgreSQL is in SSL mode in *HCL Workload Automation: Planning and Installation*.

You can run the `configureDb` command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

Default values are stored in the `configureDb<database_vendor>.properties` file, located in `image_location/TWS/interp_name`. For an example of a properties file, see [What is the content of a database properties file? on page 96](#).

If you need to modify any of the default values, edit the `configureDb<database_vendor>.properties` file, but do not modify the `configureDb<database_vendor>.template` file located in the same path.

To create and populate the HCL Workload Automation database and tables, perform the following steps:

1. On the workstation where you plan to install the master domain manager, extract the HCL Workload Automation package to a directory of your choice.
2. Browse to the `image_location/TWS/interp_name/Tivoli_MDM_interp_name/TWS/tws_tools` path.
3. Edit the `create_database.sql` file by replacing the default value for the database name (**TWS**) with the name you intend to use.
4. Provide the `create_database.sql` file to the DB2 administrator to run on the DB2 database.

The following command creates the HCL Workload Automation database:

```
db2 -tvf file_location/create_database.sql
```

5. Instruct the DB2 administrator to create the DB2 user on the server hosting the DB2 database. You will then specify this user with the `dbuser` parameter when creating and populating the database with the `configureDb` command on the master domain manager.
6. Browse to the path `image_location/TWS/interp_name`.
7. Type the following command to create and populate the HCL Workload Automation database tables with typical settings:

On Windows operating systems

```
cscript configureDb.vbs --rdbmstype DB2 --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser DB_administrator
--dbadminuserpw DB_administrator_password
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype DB2 --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser DB_administrator
--dbadminuserpw DB_administrator_password
```

where:

--rdbmstype

The database vendor.

--dbhostname *db_hostname*

The host name or IP address of database server.

--dbport *db_port*

The port of the database server.

--dbname *db_name*

The name of the HCL Workload Automation database.

--dbuser *db_user*

The database user you must create before running the `configureDb` command.

--dbadminuser *db_admin_user*

The database administrator user that creates the HCL Workload Automation schema objects on the database server.

--dbadminuserpw *db_admin_password*

The password of the DB administrator user that creates the HCL Workload Automation schema objects on the database server. Special characters are not supported.



Note: The following parameters are also required when installing the master components and their values must be the same:

- **--rdbmstype**
- **--dbhostname**
- **--dbport**
- **--dbname**
- **--dbuser**

Results

You have now successfully created and populated the HCL Workload Automation database.

What to do next

You can now proceed to [Creating and populating the database for DB2 for the Dynamic Workload Console on page 63](#).

Creating and populating the database for DB2 for the Dynamic Workload Console

Instructions for creating and populating the Dynamic Workload Console database for DB2

Before you begin

Ensure a DB2 database is installed.

About this task



You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in the section about [FAQ - Database customizations](#) in *Planning and Installation*.

DB2 requires a specific procedure in which you first create the database and then create and populate the database tables. To simplify the database creation, a customized SQL file named `create_database.sql` is provided containing the specifics for creating the Dynamic Workload Console database. The database administrator can use this file to create the database. After the database has been created, you can proceed to create and populate the database tables.

You can optionally configure DB2 in SSL mode on UNIX operating systems by specifying the `sslkeysfolder` and `sslpassword` parameters when you run the `configureDb` command. For more information, see the topic about using certificates when DB2 or PostgreSQL is in SSL mode in *HCL Workload Automation: Planning and Installation*.

You can run the `configureDb` command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

Default values are stored in the `configureDb.properties` file, located in `image_location`. If you need to modify any of the default values, edit the `configureDb.properties` file, but do not modify the `configureDb.template` file located in the same path.

For an example of a properties file, see [What is the content of a database properties file? on page 96](#).

To create and populate the Dynamic Workload Console database and schema for DB2, perform the following steps:

1. On the workstation where you plan to install the Dynamic Workload Console, extract the Dynamic Workload Console package to a directory of your choice.
2. Browse to the `image_location/DWC_interp_name/tools` path.
3. Edit the `create_database.sql` file by replacing the default value for the database name (**DWC**) with the name you intend to use.
4. Provide the `create_database.sql` file to the DB2 administrator to run on the DB2 database.

The following command creates the Dynamic Workload Console database:

```
db2 -tvf <file_location>/create_database.sql
```

5. Instruct the DB2 administrator to create the DB2 user on the server hosting the DB2 database. You will then specify this user with the `dbuser` parameter when creating and populating the database with the `configureDb` command on the Dynamic Workload Console. When you run the `configureDb` command, this user is automatically granted access to the Dynamic Workload Console tables on the database server.
6. On the server where you plan to install the Dynamic Workload Console, browse to `image_location/DWC_interp_name`.

7. Type the following command to create and populate the Dynamic Workload Console database tables with typical settings:

On Windows operating systems

```
cscript configureDb.vbs --rdbmstype DB2 --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype DB2 --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
```

where:

--rdbmstype

The database vendor.

--dbhostname *db_hostname*

The host name or IP address of database server.

--dbport *db_port*

The port of the database server.

--dbname *db_name*

The name of the Dynamic Workload Console database.

--dbuser *db_user*

The database user you must create before running the configureDb command. When you run the configureDb command, this user is automatically granted access to the Dynamic Workload Console tables on the database server.

--dbadminuser *db_admin_user*

The database administrator user that creates the Dynamic Workload Console schema objects on the database server.

--dbadminuserpw *db_admin_password*

The password of the DB administrator user that creates the Dynamic Workload Console schema objects on the database server. Special characters are not supported.



Note: The following parameters specified with the configureDb command are also required when installing the Dynamic Workload Console and their values must be the same:

- **--rdbmstype**
- **--dbhostname**
- **--dbport**



- **--dbname**
- **--dbuser**

Results

You have now successfully created and populated the Dynamic Workload Console database.

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

What to do next

You can now proceed to [Creating the HCL Workload Automation administrative user on page 99](#).

Creating and populating the database for DB2 for z/OS for the Dynamic Workload Console

Instructions for creating and populating the database for DB2 for z/OS for Dynamic Workload Console

Before you begin

Ensure a DB2 for z/OS database is installed.

About this task



You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in the section about FAQ - Database customizations in *HCL Workload Automation: Planning and Installation*.

DB2 for z/OS requires a specific procedure in which you first create the database and then create and populate the database tables. To simplify the database creation, a sample JCL named `EQQINDWC` is provided with Package `HWAZ_950_APAR_HC00001` containing the specifics for creating the Dynamic Workload Console database. The database administrator can use this file to create the database. After the database has been created, you can proceed to create and populate the database tables.

You can run the **configureDb** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

If you need to modify any of the default values, edit the `configureDb<database_vendor>.properties` file, but do not modify the `configureDb<database_vendor>.template` file located in the same path.

To create and populate the Dynamic Workload Console database and schema for DB2 for z/OS, perform the following steps:

1. From the SEQQSAMP library, edit the `EQQINDWC` sample JCL as required.



Note: The `EQQINDWC` sample JCL is provided with the Package `HWAZ_950_APAR_HC00001`. If you did not install this APAR, create a JCL named `EQQINDWC` that looks like the following example:

```
//JOB CARD
//*****
//*
//* SECURITY CLASSIFICATION:
//* Licensed Materials - Property of HCL 5698-T08
//* Copyright HCL Technologies Ltd. 2020 All rights reserved.
//* US Government Users Restricted Rights - Use, duplication
//* or disclosure restricted by GSA ADP Schedule Contract
//*
//*
//* CREATES DB2 STORAGE GROUP AND DATABASE for DWC
//* NOTE1: You must tailor this JCL sample to conform to
//* installation standards defined at your location.
//* - Add a JOB card
//* - Change following DB/2 values according to your
//* current environment:
//* - DSN.V11R1M0.SDSNLOAD DB/2 library
//* - DSN111.RUNLIB.LOAD DB/2 run library
//* - DBB1 DB/2 system name
//* - DSNTIA11 DB/2 DSNTIAD plan name
//* - volname volume name
//* - catname catalog name
//* - Change all the occurrences of
//* TWSSDWC if you need a storage group with a different name
//*
//* Flag Reason Rlse Date Origin Flag Description
//* -----
//* $EGE=PH22448 950 200121 ZLIB: DB2 on zLiberty
//* $ETA=PH53936 101 220418 MR: EQQINDWC MEMBER OF SEQQSAMP FOR
//* CREATION OF DB2 DATABASE FOR
//* DWCFails FOR DB2 V12R1M504 OR
//* higher levels
//*****
//EQQINDWC EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DISP=SHR,DSN=DSN.V11R1M0.SDSNLOAD
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
    DSN SYSTEM(DBB1)
    RUN PROGRAM(DSNTIAD) PLAN(DSNTIA11) LIB('DSN111.RUNLIB.LOAD')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
SET CURRENT APPLICATION COMPATIBILITY = 'V10R1';
CREATE STOGROUP TWSSDWC VOLUMES(volname) VCAT catname;
CREATE DATABASE DWC
BUFFERPOOL BP0
INDEXBP BP16K0
STOGROUP TWSSDWC
CCSID UNICODE;
```



```
COMMIT;
```

2. Instruct the DB2 for z/OS administrator to create the DB2 for z/OS user on the server hosting the DB2 for z/OS database. You will then specify this user with the `dbuser` parameter when creating and populating the database with the `configureDb` command on the Dynamic Workload Console. When you run the `configureDb` command, this user is automatically granted access to the Dynamic Workload Console tables on the database server.
3. On the server where you plan to install the Dynamic Workload Console, browse to the directory where you extracted the Dynamic Workload Console image.
4. Type the following command to create and populate the Dynamic Workload Console database tables with typical settings:

On Windows operating systems

```
cscript configureDb.vbs --rdbmstype DB2Z --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
--zlocationname zOS_location_containing_db --zbufferpoolname buffer_pool_in_zOS_location
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype DB2Z --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
--zlocationname zOS_location_containing_db --zbufferpoolname buffer_pool_in_zOS_location
```

On z/OS operating systems

```
./configureDb.sh --rdbmstype DB2Z --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
--zlocationname zOS_location_containing_db --zbufferpoolname buffer_pool_in_zOS_location
```

where:

--rdbmstype

The database vendor.

--dbhostname *db_hostname*

The host name or IP address of database server.

--dbport *db_port*

The port of the database server.

--dbname *db_name*

The name of the Dynamic Workload Console database.

--dbuser *db_user*

The database user you must create before running the `configureDb` command. When you run the `configureDb` command, this user is automatically granted access to the Dynamic Workload Console tables on the database server.

--dbadminuser *db_admin_user*

The database administrator user that creates the Dynamic Workload Console schema objects on the database server.

--dbadminuserpw *db_admin_password*

The password of the DB administrator user that creates the Dynamic Workload Console schema objects on the database server. Special characters are not supported.

--zlocationname *zos_location_containing_db*

The name of an already existing location in the z/OS environment that will contain the new database. The default value is LOC1.

--zbufferpoolname *buffer_pool_in_zos_location*

The name of an already existing buffer pool created in the location specified by `--zlocationname`. The default value is BP32K.



Note: The following parameters specified with the **configureDb** command are also required when installing the Dynamic Workload Console and their values must be the same:

- **--rdbmstype**
- **--dbhostname**
- **--dbport**
- **--dbname**
- **--dbuser**
- **--zlocationname**

Results

You have now successfully created and populated the Dynamic Workload Console database.

What to do next

You can now proceed to [Creating the HCL Workload Automation administrative user on page 99](#).

Creating the database for Oracle and Amazon RDS for Oracle for the master domain manager

Instructions for creating and populating the HCL Workload Automation database for Oracle and Amazon RDS for Oracle for the master domain manager

Before you begin

Ensure the following required tablespaces have been already created on the Oracle database server which hosts the master domain manager database:

- tablespace for HCL Workload Automation data
- tablespace for HCL Workload Automation log
- tablespace for HCL Workload Automation plan

About this task



You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in [FAQ - Database customizations on page 87](#).

You can run the `configureDb` command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

Default values are stored in the `configureDbOracle.properties` file, located in `image_location/TWS/interp_name`.

If you need to modify any of the default values, edit the `configureDbOracle.properties` file, but do not modify the `configureDbOracle.template` file located in the same path. For an example of a properties file, see [What is the content of a database properties file? on page 96](#).

To create and populate the HCL Workload Automation database and schema, perform the following steps:

1. On the server where you plan to install the master domain manager, extract the HCL Workload Automation package to a directory of your choice.
2. Browse to `image_location/TWS/interp_name`.
3. Type the following command to create and populate the HCL Workload Automation database with typical settings:

On Windows operating systems

```
cscript configureDb.vbs --rdmstype ORACLE --dbname service_name
--dbuser db_user --dbpassword DB_password --dbhostname DB_hostname
--dbadminuser DB_administrator --dbadminuserpw DB_administrator_password
--iwsname USERS --iwslogtsname USERS --iwsplantsname USERS
```

On UNIX operating systems

```
./configureDb.sh --rdmstype ORACLE --dbname service_name
--dbuser db_user --dbpassword DB_password --dbhostname DB_hostname
--dbadminuser DB_administrator --dbadminuserpw DB_administrator_password
--iwsname USERS --iwslogtsname USERS --iwsplantsname USERS
```

where:

--rdmstype

The database vendor.

--dbhostname *db_hostname*

The host name or IP address of database server.

--dbport *db_port*

The port of the database server.

--dbname *db_name*

The service name of the HCL Workload Automation database.

dbuser *db_user*

The user to be granted access to the HCL Workload Automation tables on the database server.

--dbpassword *db_password*

The password for the user that has been granted access to the HCL Workload Automation tables on the database server. Special characters are not supported.

--dbadminuser *db_admin_user*

The database administrator user that creates the HCL Workload Automation schema objects on the database server.

--dbadminuserpw *db_admin_password*

The password of the DB administrator user that creates the HCL Workload Automation schema objects on the database server. Special characters are not supported.

--iwststname|-tn *table_space_name*

The name of the tablespace for HCL Workload Automation data. This parameter is required.

--iwslogstname|-ln *log_table_space*

The name of the tablespace for HCL Workload Automation log. This parameter is required.

--iwsplantsname|-pn *plan_table_space*

The name of the tablespace for HCL Workload Automation plan. This parameter is required.



Note: The following parameters specified with the `configureDb` command are also required when installing the server components and their values must be the same:

- **rdbmstype**
- **dbhostname**
- **dbport**
- **dbname**
- **dbuser**
- **dbpassword**

Results

You have now successfully created and populated the HCL Workload Automation database.

You can now proceed to [Creating the database for Oracle and Amazon RDS for Oracle for the Dynamic Workload Console on page 72](#).

Creating the database for Oracle and Amazon RDS for Oracle for the Dynamic Workload Console

Instructions for creating and populating the Dynamic Workload Console database for Oracle and Amazon RDS for Oracle

Before you begin

Ensure the required tablespace for Dynamic Workload Console data has been already created on the Oracle database server which hosts the Dynamic Workload Console database.

About this task



You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in [FAQ - Database customizations on page 87](#).

You can run the `configureDb` command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

Default values are stored in the `configureDbOracle.properties` file, located in `image_location`. If you need to modify any of the default values, edit the `configureDbOracle.properties` file, but do not modify the `configureDbOracle.template` file located in the same path.

To create and populate the Dynamic Workload Console database, perform the following steps:

1. On the server where you plan to install the Dynamic Workload Console, extract the Dynamic Workload Console package to a directory of your choice.
2. Browse to the directory where you extracted the package.
3. Type the following command to populate the Dynamic Workload Console database with typical settings:

On Windows operating systems

```
cscript configureDb.vbs --rdmstype ORACLE --dbname service_name
--dbuser db_user --dbpassword DB_password --dbhostname DB_hostname
--dbadminuser DB_administrator --dbadminuserpw DB_administrator_password
--iwstname USERS
```

On UNIX operating systems

```
./configureDb.sh --rdmstype ORACLE --dbname service_name
--dbuser db_user --dbpassword DB_password --dbhostname DB_hostname
--dbadminuser DB_administrator --dbadminuserpw DB_administrator_password
--iwstname USERS
```

where:

--rdbmstype

The database vendor.

--dbname *db_name*

The service name of the Dynamic Workload Console database.

dbuser *db_user*

The user to be granted access to the Dynamic Workload Console tables on the database server.

--dbpassword *db_password*

The password for the user that has been granted access to the Dynamic Workload Console tables on the database server. Special characters are not supported.

--dbhostname *db_hostname*

The host name or IP address of database server.

--dbadminuser *db_admin_user*

The database administrator user that creates the Dynamic Workload Console schema objects on the database server.

--dbadminuserpw *db_admin_password*

The password of the DB administrator user that creates the Dynamic Workload Console schema objects on the database server. Special characters are not supported.

--iwststname|-tn *table_space_name*

The name of the tablespace for Dynamic Workload Console data. This parameter is required.



Note: The following parameters specified with the `configureDb` command are also required when installing the Dynamic Workload Console and their values must be the same:

- **rdbmstype**
- **dbhostname**
- **dbport**
- **dbname**
- **dbuser**
- **dbpassword**

Results

You have now successfully created and populated the Dynamic Workload Console database.

What to do next

You can now proceed to [Creating the HCL Workload Automation administrative user on page 99](#).

Creating the database for MSSQL for the master domain manager

Instructions for creating and populating the HCL Workload Automation database for MSSQL for the master domain manager

About this task



You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in [FAQ - Database customizations on page 87](#).

You can run the `configureDb` command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. By default, MSSQL authentication is used. To modify the authentication type, see [How can I specify the authentication type when using an MSSQL database? on page 93](#).

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

Default values are stored in the `configureDbMSSQL.properties` file, located in `image_location/TWS/interp_name`.

If you need to modify any of the default values, edit the `configureDbMSSQL.properties` file, but do not modify the `configureDbMSSQL.template` file located in the same path. For an example of a properties file, see [What is the content of a database properties file? on page 96](#).



Note: Only on Windows systems hosting an MSSQL database, the path hosting the tablespaces must be existing before you run the `configureDb.vbs` command.

To create the HCL Workload Automation database and schema, perform the following steps:

1. Only on Windows systems hosting an MSSQL database, create the path for hosting the following tablespaces, if the path is not already existing:
 - TWS_DATA
 - TWS_LOG
 - TWS_PLAN
2. Only on Windows systems hosting an MSSQL database, specify the path for the tablespaces when running the `configureDb.vbs` command or when filling in the `configureDbMSSQL.properties` properties file with the following parameters:
 - `-iwtspath`
 - `-iwslogtspath`
 - `-iwsplantspath`
3. On the server where you plan to install the master domain manager, extract the HCL Workload Automation package to a directory of your choice.
4. Browse to `image_location/TWS/interp_name`.
5. To populate the HCL Workload Automation database with typical settings, type the following command:

On Windows operating systems

```
cscript configureDb.vbs --rdbmstype MSSQL --dbname db_name
--dbhostname db_hostname --dbadminuser db_administrator
--dbadminuserpw db_administrator_password
--iwstspath DATA_tablespace_path
--iwslogtspace LOG_tablespace_path
--iwsplantspace PLAN_tablespace_path
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype MSSQL --dbname db_name
--dbhostname db_hostname --dbadminuser db_administrator
--dbadminuserpw DB_administrator_password
--iwstspath DATA_tablespace_path
--iwslogtspace LOG_tablespace_path
--iwsplantspace PLAN_tablespace_path
```

where:

--rdbmstype

The database vendor.

--dbhostname db_hostname

The host name or IP address of database server.

--dbport db_port

The port of the database server.

--dbname db_name

The name of the HCL Workload Automation database.

dbuser db_user

The user to be granted access to the HCL Workload Automation tables on the database server.

--dbadminuser db_admin_user

The database administrator user that creates the HCL Workload Automation schema objects on the database server.

--dbadminuserpw db_admin_password

The password of the DB administrator user that creates the HCL Workload Automation schema objects on the database server. Special characters are not supported.

--iwstspath|-tp table_space

The path of the tablespace for HCL Workload Automation or Dynamic Workload Console data. This parameter is optional. The default value for all databases other than Oracle is:

For all operating systems, except z/OS

TWS_DATA

For z/OS operating system**TWSDATA**

Only on Windows systems hosting an MSSQL database, ensure the folder for the tablespace is already existing before running the configureDb command and specify the path using this parameter. Specify the path using forward slashes (/), for example: `c : / <my_path> / TWS_DATA`.

--iwslogtspath|-lp log_path_table_space

The path of the tablespace for HCL Workload Automation log. This parameter is optional. The default value for all databases other than Oracle is **TWS_LOG**. This parameter applies only to the server components. Only on Windows systems hosting an MSSQL database, ensure the folder for the tablespace is already existing before running the configureDb command and specify the path using this parameter. Specify the path using forward slashes (/), for example: `c : / <my_path> / TWS_LOG`.

--iwsplantspath|-pp plan_path_table_space

The path of the tablespace for HCL Workload Automation plan. This parameter is optional. The default value for all databases other than Oracle is **TWS_PLAN**. This parameter applies only to the server components.

Only on Windows systems hosting an MSSQL database, ensure that the folder for the tablespace is already existing before running the configureDb command and specify the path using this parameter. Specify the path using forward slashes (/), for example: `c : / <my_path> / TWS_PLAN`.



Note: The following parameters specified with the configureDb command are also required when installing the server components and their values must be the same:

- **rdbmstype**
- **dbhostname**
- **dbport**
- **dbname**
- **dbuser**

Results

You have now successfully created and populated the HCL Workload Automation database.

You can now proceed to [Creating and populating the database for MSSQL for the Dynamic Workload Console on page 76](#).

Creating and populating the database for MSSQL for the Dynamic Workload Console

Instructions for creating and populating the Dynamic Workload Console database for MSSQL

About this task

You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in [FAQ - Database customizations on page 87](#).

You can run the `configureDb` command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. By default, MSSQL authentication is used. To modify the authentication type, see [How can I specify the authentication type when using an MSSQL database? on page 93](#).

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#). If you need to modify any of the default values, edit the `configureDbMSSQL.properties` file, but do not modify the `configureDbMSSQL.template` file located in the same path. For an example of a properties file, see [What is the content of a database properties file? on page 96](#).

Default values are stored in the `configureDbMSSQL.properties` file, located in `image_location`.



Note: Only on Windows systems hosting an MSSQL database, the path hosting the tablespace must be existing before you run the `configureDb.vbs` command.

To create the Dynamic Workload Console database and schema, perform the following steps:

1. Only on Windows systems hosting an MSSQL database, create the path for hosting the following tablespace, if the path is not already existing:
 - TWS_DATA
2. Only on Windows systems hosting an MSSQL database, specify the path to the folder when running the `configureDb.vbs` command or when filling in the `configureDbMSSQL.properties` properties file with the following parameter:
 - `--iwstspath`
3. On the server where you plan to install the Dynamic Workload Console, extract the Dynamic Workload Console package to a directory of your choice.
4. To populate the Dynamic Workload Console database with typical settings, type the following command:

On Windows operating systems

```
cscript configureDb.vbs --rdbmstype MSSQL --dbname db_name
--dbhostname db_hostname --dbadminuser db_administrator
--dbadminuserpw db_administrator_password
--iwstspath DATA_tablespace_path
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype MSSQL --dbname db_name
--dbhostname db_hostname --dbadminuser db_administrator
--dbadminuserpw db_administrator_password
--iwstspath DATA_tablespace_path
```

where:

--rdbmstype

The database vendor.

--dbname *db_name*

The name of the Dynamic Workload Console database.

--dbhostname *db_hostname*

The host name or IP address of database server.

--dbadminuser *db_admin_user*

The database administrator user that creates the Dynamic Workload Console schema objects on the database server.

--dbadminuserpw *db_admin_password*

The password of the DB administrator user that creates the Dynamic Workload Console schema objects on the database server. Special characters are not supported.

--iwstspath|-tp *table_space*

The path of the tablespace for HCL Workload Automation or Dynamic Workload Console data. This parameter is optional. The default value for all databases other than Oracle is:

For all operating systems, except z/OS

TWS_DATA

For z/OS operating system

TWSDATA

Only on Windows systems hosting an MSSQL database, ensure the folder for the tablespace is already existing before running the configureDb command and specify the path using this parameter. Specify the path using forward slashes (/), for example: `c:/<my_path>/TWS_DATA`.



Note: The following parameters specified with the configureDb command are also required when installing the Dynamic Workload Console and their values must be the same:

- **rdbmstype**
- **dbhostname**
- **dbport**
- **dbname**

When **--rdbmstype** is set to `MSSQL`, the default value is **sa**. To install a Dynamic Workload Console with a user different from **sa**, you must create a new user in `MSSQL` and grant all the required permissions before running the configureDb command.

Results

You have now successfully created and populated the Dynamic Workload Console database.

What to do next

You can now proceed to [Installing the Dynamic Workload Console on page 334](#).

Creating the database for MSSQL cloud-based databases for the master domain manager

Instructions for creating and populating the HCL Workload Automation database for MSSQL cloud-based databases for the master domain manager.

About this task



MSSQL cloud-based databases include the following:

- Azure SQL
- Google Cloud SQL for SQL server
- Amazon RDS for MSSQL

You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in [FAQ - Database customizations on page 87](#).

You can run the `configureDb` command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

Default values are stored in the `configureDbMSSQL.properties` file, located in `image_location/TWS/interp_name`.

If you need to modify any of the default values, edit the `configureDbMSSQL.properties` file, but do not modify the `configureDbMSSQL.template` file located in the same path. For an example of a properties file, see [What is the content of a database properties file? on page 96](#).

To create the HCL Workload Automation database and schema, perform the following steps:

1. Specify the path for the tablespaces when running the `configureDb` command or when filling in the `configureDbMSSQL.properties` properties file with the following parameters:
 - `-iwsname PRIMARY`
 - `-iwslogtsname PRIMARY`
 - `-iwsplantsname PRIMARY`
2. On the server where you plan to install the master domain manager, extract the HCL Workload Automation package to a directory of your choice.
3. Browse to `image_location/TWS/interp_name`.
4. To populate the HCL Workload Automation database with typical settings, type the following command:

On Windows operating systems

```
cscript configureDb.vbs --rdbmstype MSSQL --dbname db_name
--dbhostname db_hostname --dbadminuser db_administrator
--dbadminuserpw db_administrator_password
--iwststname PRIMARY
--iwslogtsname PRIMARY
--iwsplantsname PRIMARY
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype MSSQL --dbname db_name
--dbhostname db_hostname --dbadminuser db_administrator
--dbadminuserpw DB_administrator_password
--iwststname PRIMARY
--iwslogtsname PRIMARY
--iwsplantsname PRIMARY
```

where:

--rdbmstype

The database vendor.

--dbhostname *db_hostname*

The host name or IP address of database server.

--dbname *db_name*

The name of the HCL Workload Automation database.

dbuser *db_user*

The user to be granted access to the HCL Workload Automation tables on the database server.

--dbadminuser *db_admin_user*

The database administrator user that creates the HCL Workload Automation schema objects on the database server.

--dbadminuserpw *db_admin_password*

The password of the DB administrator user that creates the HCL Workload Automation schema objects on the database server. Special characters are not supported.

--iwststname|-tn *table_space_name*

The name of the tablespace for HCL Workload Automation data. The default value is PRIMARY. Do not modify this value.

--iwslogtsname|-ln *log_table_space*

The name of the tablespace for HCL Workload Automation log. The default value is PRIMARY. Do not modify this value.

--iwsplantsname|-pn *plan_table_space*

The name of the tablespace for HCL Workload Automation plan. The default value is PRIMARY. Do not modify this value.

Results

You have now successfully created and populated the HCL Workload Automation database.

You can now proceed to [Creating and populating the database for MSSQL cloud-based databases for the Dynamic Workload Console on page 81](#).

Creating and populating the database for MSSQL cloud-based databases for the Dynamic Workload Console

Instructions for creating and populating the Dynamic Workload Console database for MSSQL cloud-based databases

About this task



MSSQL cloud-based databases include the following:

- Azure SQL
- Google Cloud SQL for SQL server
- Amazon RDS for MSSQL

You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in [FAQ - Database customizations on page 87](#).

You can run the `configureDb` command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

If you need to modify any of the default values, edit the `configureDbMSSQL.properties` file, but do not modify the `configureDbMSSQL.template` file located in the same path. Default values are stored in the `configureDbMSSQL.properties` file, located in `image_location`.

For an example of a properties file, see [What is the content of a database properties file? on page 96](#).

To create the Dynamic Workload Console database and schema, perform the following steps:

1. Specify the path to the folder when running the `configureDb` command or when filling in the `configureDbMSSQL.properties` properties file with the following parameter:
 - `--iwstname PRIMARY`
2. On the server where you plan to install the Dynamic Workload Console, extract the Dynamic Workload Console package to a directory of your choice.
3. To populate the Dynamic Workload Console database with typical settings, type the following command:

On Windows operating systems

```
cscript configureDb.vbs --rdbmstype MSSQL --dbname db_name
--dbhostname db_hostname --dbadminuser db_administrator
--dbadminuserpw db_administrator_password
--iwstname PRIMARY
```

iwstname *DATA_tablespace_name*

The name of the tablespace for Dynamic Workload Console data. The default value is PRIMARY. Do not modify this value.



Note: The following parameters specified with the **configureDb** command are also required when installing the Dynamic Workload Console and their values must be the same:

- **rdbmstype**
- **dbhostname**
- **dbport**
- **dbname**
- **dbuser**

Results

You have now successfully created and populated the Dynamic Workload Console database.

What to do next

You can now proceed to [Creating the HCL Workload Automation administrative user on page 99](#).

Creating and populating the database for PostgreSQL or PostgreSQL cloud-based databases for the master domain manager

Instructions for creating and populating the HCL Workload Automation database for PostgreSQL for the master domain manager.

Before you begin

Ensure you have performed the following tasks:

- Create the PostgreSQL database and ensure it is configured to allow remote connections. To create the database, use the following command:

```
create database <database_name> with lc_collate='C' template=template0;
```

This command creates the database with the collation feature enabled.

- Create a user dedicated specifically to the new database schema and do not use the administrator user (`postgres`) for this purpose.

For more information about allowing remote connections and creating users, see the PostgreSQL documentation.

About this task



PostgreSQL cloud-based databases include the following:

- Amazon RDS for Postgres
- Google Cloud SQL for Postgres

You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in [FAQ - Database customizations on page 87](#).

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

You can optionally configure PostgreSQL in SSL mode on UNIX operating systems by specifying the `sslkeysfolder` and `sslpassword` parameters when you run the `configureDb` command. For more information, see [How can I use certificates when Db2 or PostgreSQL is in SSL mode? on page 95](#)

You can run the `configureDb` command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

Default values are stored in the `configureDbPostgresql.properties` file, located in `image_location`. If you need to modify any of the default values, edit the `configureDbPostgresql.properties` file, but do not modify the `configureDbPostgresql.template` file located in the same path. For an example of a properties file, see [What is the content of a database properties file? on page 96](#).

If you need to modify any of the default values, edit the `configureDbPostgresql.properties` file, but do not modify the `configureDbPostgresql.template` file located in the same path.

To create and populate the HCL Workload Automation database and tables, perform the following steps:

1. On the workstation where you plan to install the master domain manager, extract the HCL Workload Automation package to a directory of your choice.
2. Browse to the path `image_location/TWS/interp_name`.
3. Type the following command to create and populate the HCL Workload Automation database tables with typical settings:

On Windows operating systems

```
cscript configureDb.vbs --rdbmstype POSTGRESQL --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser DB_administrator
--dbadminuserpw DB_administrator_password
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype POSTGRESQL --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
```

```
--dbadminuser DB_administrator
--dbadminuserpw DB_administrator_password
```

where:

--rdbmstype

The database vendor.

--dbhostname *db_hostname*

The host name or IP address of database server.

--dbport *db_port*

The port of the database server.

--dbname *db_name*

The name of the HCL Workload Automation database.

--dbuser *db_user*

The database user you must create before running the `configureDb` command. Hyphens are not supported.

--dbadminuser *db_admin_user*

The database administrator user that creates the HCL Workload Automation schema objects on the database server.

--dbadminuserpw *db_admin_password*

The password of the DB administrator user that creates the HCL Workload Automation schema objects on the database server. Special characters are not supported.



Note: The following parameters are also required when installing the master components and their values must be the same:

- **--rdbmstype**
- **--dbhostname**
- **--dbport**
- **--dbname**
- **--dbuser**

Results

You have now successfully created and populated the HCL Workload Automation database.

What to do next

You can now proceed to [Creating and populating the database for PostgreSQL or PostgreSQL cloud-based databases for the Dynamic Workload Console on page 85](#).

Creating and populating the database for PostgreSQL or PostgreSQL cloud-based databases for the Dynamic Workload Console

Instructions for creating and populating the Dynamic Workload Console database for PostgreSQL

Before you begin

Ensure you have performed the following tasks:

- Create the PostgreSQL database and ensure it is configured to allow remote connections. To create the database, use the following command:

```
create database <database_name> with lc_collate='C' template=template0;
```

This command creates the database with the collation feature enabled.

- Create a user dedicated specifically to the new database schema and do not use the administrator user (`postgres`) for this purpose.

For more information about allowing remote connections and creating users, see the PostgreSQL documentation.

About this task



PostgreSQL cloud-based databases include the following:

- Amazon RDS for Postgres
- Google Cloud SQL for Postgres

You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in [FAQ - Database customizations on page 87](#).

You can optionally configure PostgreSQL in SSL mode on UNIX operating systems by specifying the `sslkeysfolder` and `sslpassword` parameters when you run the `configureDb` command. For more information, see [How can I use certificates when Db2 or PostgreSQL is in SSL mode? on page 95](#)

You can run the `configureDb` command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

Default values are stored in the `configureDbPostgresql.properties` file, located in `image_location`. If you need to modify any of the default values, edit the `configureDbPostgresql.properties` file, but do not modify the `configureDbPostgresql.template` file located in the same path. For an example of a properties file, see [What is the content of a database properties file? on page 96](#).

To create and populate the Dynamic Workload Console database and schema for PostgreSQL, perform the following steps:

1. On the workstation where you plan to install the Dynamic Workload Console, extract the Dynamic Workload Console package to a directory of your choice.
2. Browse to the path *image_location/TWS/interp_name*.
3. Type the following command to create and populate the Dynamic Workload Console database tables with typical settings:

On Windows operating systems

```
cscript configureDb.vbs --rdbmstype POSTGRESQL --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype POSTGRESQL --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
```

where:

--rdbmstype

The database vendor.

--dbhostname *db_hostname*

The host name or IP address of database server.

--dbport *db_port*

The port of the database server.

--dbname *db_name*

The name of the Dynamic Workload Console database.

--dbuser *db_user*

The database user you must create before running the configureDb command. Hyphens are not supported. When you run the configureDb command, this user is automatically granted access to the Dynamic Workload Console tables on the database server.

--dbadminuser *db_admin_user*

The database administrator user that creates the Dynamic Workload Console schema objects on the database server.

--dbadminuserpw *db_admin_password*

The password of the DB administrator user that creates the Dynamic Workload Console schema objects on the database server. Special characters are not supported.



Note: The following parameters specified with the configureDb command are also required when installing the Dynamic Workload Console and their values must be the same:

- **--rdbmstype**
- **--dbhostname**



- **--dbport**
- **--dbname**
- **--dbuser**

Results

You have now successfully created and populated the Dynamic Workload Console database.

For more information about all parameters and supported values of the configureDb command, see [Database configuration - configureDb script on page 430](#).

What to do next

You can now proceed to [Creating the HCL Workload Automation administrative user on page 99](#).

FAQ - Database customizations

A list of questions and answers related to the customization of the database:

When creating and populating a database, you might have the need to customize some parameters to suit your environment.

- [How can I modify the tablespace? on page 87](#)
- [How can I avoid providing the database administrator credentials when creating the database with DB2? on page 88](#)
- [How can I configure a different temporary directory where files get downloaded? on page 89](#)
- [How can I generate the SQL files required to create the database schema? on page 90](#)
- [How can I use Oracle partitioning? on page 91](#)
- [How can I customize the Temp tablespace on Oracle? on page 92](#)
- [How can I check database consistency to avoid schema corruption? on page 92](#)
- [How can I specify the authentication type when using an MSSQL database? on page 93](#)
- [How can I customize the JDBC drivers for the database? on page 94](#)
- [How can I grant access to the database when the user installing the product is not the database administrator? on page 95](#)
- [How can I use certificates when Db2 or PostgreSQL is in SSL mode? on page 95](#)
- [What is the content of a database properties file? on page 96](#)

How can I modify the tablespace?

How can I modify the tablespace?

If you do not want to use the default tablespace name and path, you can modify them when creating and populating the database with the configureDb command.

Proceed as follows:

1. Browse to the folder containing the configureDb command. The command is located in the following path, depending on the component for which you are installing:

master domain manager

image_location>/TWS/interp_name

Dynamic Workload Console

image_location>

2. When launching the configureDb command, as explained in [Installing the master domain manager and backup master domain manager on page 100](#) and [Installing the Dynamic Workload Console servers on page 110](#), modify the following parameters as necessary:

-iwstsname|-tn *table_space_name*

The name of the tablespace for HCL Workload Automation data. This parameter is optional. The default value is **TWS_DATA**.

-iwstspath|-tp *table_space_path*

The path of the tablespace for HCL Workload Automation data. This parameter is optional. The default value is **TWS_DATA**.

-iwslogtsname|-ln *log_table_space*

The name of the tablespace for HCL Workload Automation log. This parameter is optional. The default value is **TWS_LOG**.

-iwslogtspath|-lp *log_path_table_space*

The path of the tablespace for HCL Workload Automation log. This parameter is optional. The default value is **TWS_LOG**.

-iwsplantsname|-pn *plan_table_space*

The name of the tablespace for HCL Workload Automation plan. This parameter is optional. The default value is **TWS_PLAN**.

-iwsplantspath|-pp *plan_path_table_space*

The path of the tablespace for HCL Workload Automation plan. This parameter is optional. The default value is **TWS_PLAN**.

For more information about the configureDb command, see [Database configuration - configureDb script on page 430](#).

How can I avoid providing the database administrator credentials when creating the database with DB2?

Minimum required grants to manage the HCL Workload Automation database with DB2

If you prefer to keep the database administrator credentials confidential and you are using DB2, you can assign a user a minimum set of grants to create, access, and modify the HCL Workload Automation database.

Using the `configureDb` command, you can perform the following operations:

- Create the custom SQL statement to create or upgrade the HCL Workload Automation database schema.
- Apply the generated SQL statement to upgrade the HCL Workload Automation schema to the latest version.

Each of the previous steps requires a set of minimum grants.

Minimum required grants to create the HCL Workload Automation database and table spaces

Run the `configureDb` command with the `--execsql` parameter set to **FALSE** to generate the `customSQLAdmin.sql` file containing the **CREATE DATABASE** statement.

After creating the database, run the `configureDb` command with the `--execsql` parameter set to **FALSE** to generate the `customSQL.sql` file containing the SQL statements to create table spaces and schemas. Extract from the `customSQL.sql` file the statements to **CREATE** the **BUFFERPOOLS** and **TABLESPACES**.

To create the HCL Workload Automation database and the **BUFFERPOOLS** and **TABLESPACES**, one of the following minimum grants is required:

- SYSADM
- SYSCTRL
- SELECT privilege on the PRIVILEGES administrative view

Grant to create and upgrade the HCL Workload Automation database schema

To create the HCL Workload Automation schema in the database, run the `configureDb` command with the following authorities and authorizations:

- CREATETAB on database
- CONNECT on database
- USE on all HCL Workload Automation table spaces
- SELECT privilege on the PRIVILEGES administrative view

Run the `configureDb` command with the `--execsql` parameter set to **TRUE** to create or upgrade the HCL Workload Automation database schema.

How can I configure a different temporary directory where files get downloaded?

Customizing the working directory of the database.

If you do not want to use the default working directory, where temporary files are stored, you can customize it when creating and populating the database with the `configureDb` command.

Proceed as follows:

1. Browse to the folder containing the configureDb command. The command is located in the following path, depending on the component for which you are installing:

master domain manager

image_location/TWS/interp_name

Dynamic Workload Console

image_location

2. When launching the configureDb command, as explained in [Installing the master domain manager and backup master domain manager on page 100](#) and [Installing the Dynamic Workload Console servers on page 110](#), modify the following parameter as necessary:

work_dir

The working directory where you extract the installation image. It also contains the output produced by the command, such as the SQL statements if you set the **execsql** parameter to **false**. The default value is `/tmp` on UNIX operating systems and `C:\tmp` on Windows operating systems.

For more information about the configureDb command, see [Database configuration - configureDb script on page 430](#).

How can I generate the SQL files required to create the database schema?

Generating the SQL files for the database schema

If you do not have the access rights to generate the schema in the database, you can create the required SQL files and then provide them to the database administrator. If you do have the access rights to generate the schema in the database, you might also want to generate the SQL files and review them before applying them to the database.

Proceed as follows:

1. Browse to the folder containing the configureDb command. The command is located in the following path, depending on the component for which you are installing:

master domain manager

image_location/TWS/interp_name

Dynamic Workload Console

image_location

2. When launching the configureDb command on the workstation where you plan to install the master domain manager or Dynamic Workload Console, as explained in [Creating and populating the database on page 59](#), set `- execsql` parameter set to **false**:

-execsql|-es *execute_sql*

Set to **true** to generate and run the SQL file, set to **false** to generate the SQL statement without running it. The resulting files are stored in the path defined in the **work_dir** parameter. This option is useful if you want to review the file before running it. This parameter is optional. The default value is **true**.

3. The command creates the relevant SQL scripts containing the settings you have defined in the command line. The files are created in the working directory, which by default is `/tmp` on UNIX operating systems and `C:\tmp` on Windows operating systems.

For more information about the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

How can I use Oracle partitioning?

Using Oracle partitioning.

Partitioning is a powerful functionality that enables tables, indexes, and index-organized tables to be subdivided into smaller pieces, allowing these database objects to be managed and accessed at a finer level of granularity. Moreover, the Oracle partitioning feature can improve the performance of the auditing feature and event-driven workload automation. This functionality improves rule management performance, in particular the following queries:

- `event_rule_instance`
- `action_run`
- `operator_messages`

If partitioning is already enabled in your Oracle database, proceed as follows:

1. Browse to the folder containing the `configureDb` command. The command is located in the following path, depending on the component for which you are installing:

master domain manager

`image_location/TWS/interp_name`

Dynamic Workload Console

`image_location`

2. When launching the `configureDb` command, as explained in [Installing the master domain manager and backup master domain manager on page 100](#) and [Installing the Dynamic Workload Console servers on page 110](#), modify the following parameter as necessary:

--usePartitioning

Only applies when installing the master domain manager. Set to **true** if you want to use the Oracle partitioning feature, otherwise set it to **false**. This parameter is optional. The default value is **true**.

For more information about the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

How can I customize the Temp tablespace on Oracle?

Customizing the HCL Workload Automation Temp tablespace on Oracle

If you do not want to use the default Oracle Temp tablespace, you can customize it when creating and populating the database with the configureDb command.

Proceed as follows:

1. Browse to the folder containing the configureDb command. The command is located in the following path, depending on the component for which you are installing:

master domain manager

image_location/TWS/interp_name

Dynamic Workload Console

image_location

2. When launching the configureDb command, as explained in [Installing the master domain manager and backup master domain manager on page 100](#) and [Installing the Dynamic Workload Console servers on page 110](#), modify the following parameter:

--iwsTempTsName IWS_temp_path

Only applies when installing the master domain manager. The path of the tablespace for HCL Workload Automation temporary directory. This parameter is optional. The default value is **TEMP**.

For more information about the configureDb command, see [Database configuration - configureDb script on page 430](#).

How can I check database consistency to avoid schema corruption?

Checking and maintaining database consistency

The database administrator can verify if the database schema has changed and repair any inconsistencies.

Proceed as follows:

1. Browse to the folder containing the configureDb command. The command is located in the following path, depending on the component for which you are installing:

master domain manager

image_location/TWS/interp_name

Dynamic Workload Console

image_location

2. When launching the configureDb command, as explained in [Installing the master domain manager and backup master domain manager on page 100](#) and [Installing the Dynamic Workload Console servers on page 110](#), set the
 - `execsql` parameter to **false**:

-execsql|-es *execute_sql*

Set to **true** to generate and run the SQL file, set to **false** to generate the SQL statement without running it. The resulting files are stored in the path defined in the **work_dir** parameter. This option is useful if you want to review the file before running it. This parameter is optional. The default value is **true**.

This parameter generates a number of SQL files, which you can check to look for any inconsistencies. For example, if you find CREATE instructions, this means that some records or indexes are missing in the database.

3. If you identify any inconsistencies, provide the files to the database administrator to run on the database and fix the inconsistencies.

For more information about the configureDb command, see [Database configuration - configureDb script on page 430](#).

How can I specify the authentication type when using an MSSQL database?

Configuring the authentication type for the MSSQL database.

When using an MSSQL database, you can choose between two different authentication types:

- MSSQL authentication. This is the default value.
- Windows authentication

To define the authentication type, proceed as follows:

1. Browse to the folder containing the configureDb command. The command is located in the following path, depending on the component for which you are installing:

master domain manager

image_location/TWS/interp_name

Dynamic Workload Console

image_location

2. When launching the configureDb command, as explained in [Creating and populating the database on page 59](#), specify the **auth_type** argument with one of the following values:

SQLSERVER

Enables MSSQL authentication type. Only the user specified with the **--dbadminuser** argument has the grants to administer the HCL Workload Automation database. This is the default value.

WINDOWS

Enables Windows authentication type. The Windows user you used to log on to the workstation is assigned the grants to administer the HCL Workload Automation database.

For more information about all parameters and supported values of the configureDb command, see [Database configuration - configureDb script on page 430](#).

How can I customize the JDBC drivers for the database?

How can I customize the JDBC drivers for the database?

If you do not want to use the default JDBC drivers, for example because more updated drivers have been released in the meantime, you can replace them with a few easy steps for both the master domain manager and Dynamic Workload Console.

Proceed as follows:

1. Download the updated JDBC drivers for your database.
2. Create a backup of the existing JDBC drivers installed together with the product in the following paths:

master domain manager

On Windows operating systems

`TWA_home\TWS\jdbcdrivers\default_RDBMS`

On UNIX operating systems

`TWA_home/TWS/jdbcdrivers/default_RDBMS`

Dynamic Workload Console

On Windows operating systems

`DWC_home\jdbcdrivers\default_RDBMS`

On UNIX operating systems

`DWC_home/jdbcdrivers/default_RDBMS`

where

default_RDBMS

Indicates one of the following directories related to the database you are using for the master domain manager and the Dynamic Workload Console:

- db2
- db2z
- mssql (applies to MSSQL and supported MSSQL cloud-based databases)
- oracle (applies to Oracle and Amazon RDS for Oracle)
- postgresql

3. Stop Open Liberty for master domain manager and Dynamic Workload Console, as described in the section about Application server - starting and stopping in *Administration Guide*.
4. Replace the default JDBC drivers with the updated ones. Ensure you maintain the same path and rename the updated drivers to the exact name of the previous drivers.
5. Start Open Liberty for master domain manager and Dynamic Workload Console, as described in the section about Application server - starting and stopping in *Administration Guide*.



Note: When you upgrade the master domain manager and Dynamic Workload Console to a new product version, the customized JDBC drivers are replaced by the drivers included in the product installation packages. To continue using custom JDBC drivers, repeat this procedure.

How can I grant access to the database when the user installing the product is not the database administrator?

Steps to grant access to the database tables when the user installing the product is not the database administrator.

If the user installing the product is not the database administrator, ensure you run the `grant_twsuser.sql` script before you run the `configureDb` script.

This ensures the database user is granted all proper rights.

The `grant_twsuser.sql` is available in `TWA_home/TWS/dbtools/<database_vendor>/sql`.

How can I use certificates when Db2 or PostgreSQL is in SSL mode?

How can I use certificates when Db2 or PostgreSQL is in SSL mode?

If you have Db2 or PostgreSQL set up in SSL mode on a UNIX operating system, you can add the database certificate to the existing certificates. You can use this configuration on the following components:

- master domain manager
- dynamic domain manager
- Dynamic Workload Console

Proceed as follows:

1. On the workstation where you plan to install the master domain manager, create a folder for storing the certificates.
2. Within this folder, create a subfolder named `additionalCAs`.
3. Obtain the certificates from the database administrator.
4. Store the certificates in `.cert` format in the `additionalCAs` folder.
5. Log in to the component for which you are configuring the database, as listed above.
6. Run the `configureDb` script as explained in [Creating and populating the database for DB2 for the master domain manager on page 61](#) and [Creating and populating the database for DB2 for the Dynamic Workload Console on page 63](#), or in [Creating and populating the database for PostgreSQL or PostgreSQL cloud-based databases for the master domain manager on page 82](#) and [Creating and populating the database for PostgreSQL or PostgreSQL cloud-based databases for the Dynamic Workload Console on page 85](#), depending on the database you are using. Ensure you use the `sslkeyfolder` and `sslkeyfolder` parameter to specify the path to the folder containing the certificates.
7. Proceed with the installation as described in [Typical installation scenario on page 54](#).

What is the content of a database properties file?

Contents of the `configureDB.properties` file.

You can use properties files for providing input to the `configureDB` command without typing parameters in the command line when creating the database for the master domain manager and Dynamic Workload Console.

Consider the following example for the master domain manager database:

```
#This properties are the default for configureDb.sh command in configureDb.template file
#This properties are the input for configureDb.sh command with -f option in configureDb.properties file
#N.B.configureDb.template must not be changed, while configureDb.properties can be changed when using -f option

#--lang language: C|en|de|es|fr|it|ja|ko|pt_BR|ru|zh_CN|zh_TW
LANG=

#--work_dir Working directory where user has write access. Used to modify input file for the db tool (optional,
  default: see below)
WORK_DIR=

#--log_dir Working directory where user has write access. Used to log (optional, default: see below)
CONFDB_LOG_DIR=

#--rdbmstype|-r The rdbmstype:      DB2 | ORACLE | MSSQL | POSTGRESQL
RDBMS_TYPE=

#--componenttype The IWS component that must be installed: MDM, BKM, DDM or BDM (default: see below)
COMPONENT_TYPE=MDM

#--dbdriverpath
DB_DRIVER_PATH=

#--dbname The name of IWS Database (default: see below)
DB_NAME=TWS

#--dbhostname The host name or IP address of DB server
DB_HOST_NAME=

#--dbport The port of the DB server
DB_PORT=50000

#--dbadminuser DB administrator user that creates the IWS schema objects on the DB server
DB_ADMIN_USER=db2admin

#--dbadminuserpw The password of the DB administrator user that creates the IWS schema objects on the DB2
  server
DB_ADMIN_USER_PWD=

#--dbuser DB user that accesses the IWS tables on the DB server
DB_USER=db2twc

#--dbpassword DB user that accesses the IWS tables on the DB server
DB_PASSWORD=

#--wlpdir|-w wlp directory needed only if any password in input is encrypted and has the form {xor}password
WLP_INSTALL_DIR=
```

```

#--iwststname The name of the tablespace for IWS data (default: see below)
IWS_TS_NAME=TWS_DATA

#--iwstspath The path of the tablespace for IWS data (default: see below)
IWS_TS_PATH=TWS_DATA

#--iwslogstname The name of the tablespace for IWS log (default: see below)

#--iwslogspath The path of the tablespace for IWS log (default: see below)
IWS_LOG_TS_PATH=TWS_LOG

#--iwsplantsname The name of the tablespace for IWS plan (default: see below)
IWS_PLAN_TS_NAME=TWS_PLAN

#--iwsplantspath The path of the tablespace for IWS plan (default: see below)
IWS_PLAN_TS_PATH=TWS_PLAN

# Automatically apply the generated SQL statements needed to create the IWS database schema objects (Default:
TRUE)
# If you want manually apply the generated statement in ./customSQL.sql file, set FALSE.
#--execsql
EXEC_GENERATED_SQL=TRUE

# -----
# needed for SSL
# -----
# Configuration options when customized certificates are used for SSL connections:
#--sslkeysfolder          The name and path of the folder containing certificates in PEM format.
#                          This parameter is required if you set the --dbsslconnection parameter to true.
SSL_KEY_FOLDER=
#--sslpassword            If you provide PEM certificates with the --sslkeysfolder parameter, this is the
password for the certificates automatically generated by the installation program.
SSL_PASSWORD=
# -----
# needed for SSL  unix only
# -----
#--dbsslconnection        true | false  (DB2 only)
DB_SSL_CONNECTION=false

```

Consider the following example for the Db2 database for the Dynamic Workload Console:

```

#This properties are the default for configureDb.sh command in configureDb.template file
#This properties are the input  for configureDb.sh command with -f option in configureDb.propeties file
#N.B.configureDb.template must not be changed, while configureDb.propeties can be changed when using -f option

#--lang language: C|en|de|es|fr|it|ja|ko|pt_BR|ru|zh_CN|zh_TW
LANG=

#--work_dir Working directory where user has write access. Used to modify input file for the db tool (optional,
default: see below)
WORK_DIR=

#--log_dir Working directory where user has write access. Used to log (optional, default: see below)
CONFDB_LOG_DIR=

#--rdbmstype|-r The rdbmstype:          DB2 | DB2Z | ORACLE | MSSQL | POSTGRESQL

```

```

RDBMS_TYPE=DB2

#--componenttype The DWC component that must be installed: DWC
COMPONENT_TYPE=DWC

#--dbdriverpath
DB_DRIVER_PATH=

#--dbname The name of DWC Database (default: see below)
DB_NAME=TDWC

#--dbhostname The host name or IP address of DB server
DB_HOST_NAME=<my_DB_host>

#--dbport The port of the DB server
DB_PORT=50000

#--dbadminuser DB administrator user that creates the IWS schema objects on the DB server
DB_ADMIN_USER=db2admin

#--dbadminuserpw The password of the DB administrator user that creates the DWC schema objects on the DB2
server
DB_ADMIN_USER_PWD=<database_administrator_password>

#--dbuser DB user that accesses the DWC tables on the DB server
DB_USER=db2dwc

#--dbpassword DB user that accesses the DWC tables on the DB server
DB_PASSWORD=database_password

#--wlpdir|-w wlp directory needed only if any password in input is encrypted and has the form {xor}password
WLP_INSTALL_DIR=<WebSphere Application Server
Liberty_installation_directory>

#--iwststname The name of the tablespace for data (default: TWS_DATA)
IWS_TS_NAME=TWS_DATA

#--iwstspath The path of the tablespace for data (default: TWS_DATA)
IWS_TS_PATH=TWS_DATA

# Automatically apply the generated SQL statements needed to create the DWC database schema objects (Default:
TRUE)
# If you want manually apply the generated statement in ./customSQL.sql file, set FALSE.
#--execsql
EXEC_GENERATED_SQL=TRUE

# -----
# needed for SSL
# -----
# Configuration options when customized certificates are used for SSL connections:
#--sslkeysfolder The name and path of the folder containing certificates in PEM format.
# If you provide PEM certificates, the installation program generates the keystore and
truststore files using the password you specify with the --sslpassword parameter.
# This parameter is required if you set the --dbsslconnection parameter to true.
SSL_KEY_FOLDER=
#--sslpassword If you provide PEM certificates with the --sslkeysfolder parameter, this is the
password for the certificates automatically generated by the installation program.

```

```

SSL_PASSWORD=
# -----
# needed for SSL  unix only
# -----
#--dbsslconnection      true | false  (DB2 only)
DB_SSL_CONNECTION=false

```

Creating the HCL Workload Automation administrative user

Instructions to create the HCL Workload Automation administrative user



HCL Workload Automation administrative user

The HCL Workload Automation administrator creates the administrative user (**wauser**). The administrative user is the user for which the product will be installed in the subsequent steps. This implies that this user has full access to all scheduling objects.

The user name can contain alphanumeric, dash (-), and underscore (_) characters; it cannot contain national characters. The first character of the user name must be a letter.

The following considerations apply:

On Windows operating systems:

- If this user account does not already exist, it is automatically created at installation time.
- If installing on a Windows™ server in a domain, do not define a domain and local ID with the same user name.
- If you specify a domain user, define the name as *domain_name\user_name*.
- If you specify a local user, define the name as *system_name\user_name*. Type and confirm the password.

On UNIX and Linux operating systems:

This user account must be created manually before running the installation and must be enabled to login to the machine where the master domain manager is going to be installed. Create a user with a home directory and group. Use the appropriate UNIX and Linux operating system commands to create the user.



Important: Group names that contain a "/" (forward slash) character can cause permissions to not be set correctly. When HCL Workload Automation retrieves credentials from WebSphere Application Server Liberty, it parses the returned list of groups names assuming they are saved in the format



`<realm_name>/<group_name>`. If the group name, the realm name, or both contain a "/" character, the parsing fails.

You can also install HCL Workload Automation using a user different from the root user. This installation method is known as **no-root installation** and applies to all HCL Workload Automation components. Note that if you choose this installation method, only the user who performs the installation can use HCL Workload Automation. For this reason, the typical installation scenario described in this section uses the root user.

For more information, see [HCL Workload Automation user management on page 49](#).

Example

You can now proceed to [Installing the master domain manager and backup master domain manager on page 100](#).

Installing the master domain manager and backup master domain manager

A fresh installation for the master domain manager and the backup master domain manager

About this task



Note: Automatic failover triggers a switch to a backup master domain manager without manual intervention under certain conditions. To take advantage of this feature, you must install the master domain manager and backup master domain managers with the same user. With a fresh installation of a master domain manager on Linux and UNIX, a new extended agent is installed on the master domain manager workstation which is used to communicate where to run the FINAL job stream. For information about configuring automatic failover, see the topic about enabling automatic failover in the *Administration Guide*.

Procedure to install a master domain manager and backup master domain manager

Before you begin

Before starting the installation, ensure the following steps have been completed:

1. [Installing Open Liberty on page 56](#) on the workstation where you plan to install the master domain manager and on the workstation where you plan to install the backup master domain manager.
2. [Creating and populating the database on page 59](#)
3. [Creating the HCL Workload Automation administrative user on page 99](#)
4. Ensure you have created a license server and made a note of the server ID, which you will need when installing. For more information about enabling your product license, see [Enabling product license management on page 52](#).

5. On UNIX™ operating systems, ensure that **umask** is set to **022**. To verify that **umask** is set to the correct value, from a command prompt, run the **umask** command. If the value is different from **022**, modify it by running the following command:

```
umask 022
```



Note: When installing a backup master domain manager, the backup points to the existing HCL Workload Automation database. In this case, creating and populating the database is not required.

About this task

You can perform a typical installation, as described in the following scenario, or you can customize the installation parameters, as described in [FAQ - master domain manager and backup master domain manager customizations on page 107](#).

For more information about all **serverinst** parameters and default values, see [Server components installation - serverinst script on page 442](#).

Certificates are now required when installing or upgrading HCL Workload Automation. You can no longer install nor upgrade HCL Workload Automation without securing your environment with certificates. The required certificates are:

- ca.crt
- tls.key
- tls.crt

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

The procedure to install the master domain manager and backup master domain manager is exactly the same, with the difference that it is performed on two different workstations and that each installation points to its local Open Liberty installation. HCL Workload Automation determines whether or not a master domain manager is already present in the environment and proceeds to install a master domain manager or backup master domain manager accordingly.

The HCL Workload Automation administrator installs the master domain manager and backup master domain manager. The following information is required:

Table 3. Required information

Command parameter	Information type	Provided in..
Database information		
--rdbmstype	database type	Creating and populating the database on page 59
--dbhostname	database hostname	

Table 3. Required information

(continued)

--dbport	database port	
--dbname	database name	
--dbuser	database user name	
--dbpassword	database password	
HCL Workload Automation information		
--wouser	HCL Workload Automation administrative user name	Creating the HCL Workload Automation administrative user on page 99
--wapassword	HCL Workload Automation administrative user password	
Open Liberty information		
--wlpdir	Open Liberty installation directory	Installing Open Liberty on page 56
Security information		
sslkeysfolder	name and path of the folder containing certificates	Installing the master domain manager and backup master domain manager on page 100
--sslpassword	password for the certificates	Current procedure
HCL Workload Automation installation directory		
--inst_dir	installation directory	Current procedure
Licensing information		
--licenseserverid	The ID of the license server which processes license usage information	For more information about enabling your product license, see Enabling product license management on page 52.

You can run the **serverinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. For more information about all **serverinst** parameters and default values, see [Server components installation - serverinst script on page 442](#).

A properties file named `serverinst.properties` is available if you do not want to type parameters in the command line. This is especially useful if you need to specify many parameters or if you want to reuse the file for several installations. The file is located in `image_location/TWS/interp_name`.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

If you need to modify any of the default values, edit the `serverinst.properties` file, but do not modify the `serverinst.template` file located in the same path.

To install the master domain manager and backup master domain manager, perform the following steps:

1. Log in to the workstation where you plan to install the master domain manager.
2. Download the installation images from [HCL Software](#).
3. Browse to the folder where the `serverinst` command is located in `image_location/TWS/interp_name`.
4. Start the installation specifying a typical set of parameters. In this case, default values are used for all remaining parameters:

On Windows operating systems

```
cscript serverinst.vbs --acceptlicense yes --rdbmstype <db_type>
--dbhostname <db_hostname> --dbport <db_port> --dbname <db_name>
--dbuser <db_user> --dbpassword <db_password> --wauser <wa_user>
--wapassword <wa_password> --wlpdir <Liberty_installation_dir>\wlp
--sslkeysfolder <certificate_files_path> --sslpassword <keystore_truststore_password>
--inst_dir <installation_dir>
--licenseserverid <license_server_ID>
```

On UNIX operating systems

```
./serverinst.sh --acceptlicense yes --rdbmstype <db_type>
--dbhostname <db_hostname> --dbport <db_port> --dbname <db_name>
--dbuser <db_user> --dbpassword <db_password> --wauser <wa_user>
--wapassword <wa_password> --wlpdir <Liberty_installation_dir>/wlp
--sslkeysfolder <certificate_files_path> --sslpassword <keystore_truststore_password>
--inst_dir <installation_dir>
--licenseserverid <license_server_ID>
```

where

--acceptlicense

Specify **yes** to accept the product license.

--rdbmstype|-r rdbms_type

The database type. Supported databases are:

- **DB2**
- **ORACLE** This value applies to Oracle and Amazon RDS for Oracle
- **MSSQL** This value applies to MSSQL and MSSQL cloud-based databases.
- **POSTGRESQL**

This parameter is required and has no default value.

--dbhostname db_hostname

The host name or IP address of database server.

--dbport *db_port*

The port of the database server.

--dbname *db_name*

The name of the HCL Workload Automation database.

--dbuser *db_user*

The database user that has been granted access to the HCL Workload Automation tables on the database server.

--dbpassword *db_password*

The password for the user that has been granted access to the HCL Workload Automation tables on the database server. Special characters are not supported.

--wouser *user_name*

The user for which you are installing HCL Workload Automation.

--wapassword *wouser_password*

The password of the user for which you are installing HCL Workload Automation.

On Windows operating systems

Supported characters for the password are alphanumeric, dash (-), underscore (_), characters, and ()|?*~+.@!^

On UNIX operating systems

Supported characters for the password are any alphanumeric, dash (-), underscore (_), characters, and ()|?*~+.

--wlpdir

The path where Open Liberty is installed.

--licenseserverid

The ID of the license server which processes license usage information. This parameter is required. For more information about enabling your product license, see [Enabling product license management on page 52](#).

--sslkeysfolder *keystore_truststore_folder*

The name and path of the folder containing certificates in PEM format. The installation program automatically processes the keystore and truststore files using the password you specify with the **--sslpassword** parameter. The folder must contain the following files:

- **ca.crt**

The Certificate Authority (CA) public certificate. Note that if certificates being installed are part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate),

then this file must contain the Root CA certificate only. Any Intermediate CA certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.



Note: From V10.2.3, if certificates being installed are part of a chain, the `ca.crt` can contain also the intermediate CAs. In this case, it must begin with one or more intermediate CA certificates and end with the Root ca.

- **tls.key**

The private key of the end user certificate for the instance to be installed.

- **tls.crt**

The public part of the previous key, that is the end user certificate.

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

You can optionally create a subfolder to contain one or more `*.crt` files to be added to the server truststore as trusted CA, whose name must be `additionalCAs`. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or DB2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**. Note that if the end user certificate being installed in the instance is part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then the Intermediate CAs certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

For further information about how to generate custom certificates, see the topic about managing certificates using Certman in *HCL Workload Automation: Planning and Installation*.

--sslpassword *ssl_password*

The password for the certificates.

For more information, see [sslkeysfolder on page 450](#).

You can optionally encrypt the password using the secure script. For more information, see [Optional password encryption - secure script on page 427](#).

--inst_dir *installation_dir*

The directory of the HCL Workload Automation installation.

--licenseserverid *license_server_ID*

The ID of the license server which processes license usage information. This parameter is required.

Instructions about how to obtain the ID of the license server which processes license usage information

are provided with the mail confirming your license. For more information, see the section about License computation model in *Administration Guide* and Enabling product license management in *HCL Workload Automation: Planning and Installation*.



Note: The values for the following parameters must match the values you provided when creating and populating the database:

- **--rdbmstype**
- **--dbhostname**
- **--dbport**
- **--dbname**
- **--dbuser**
- **--dbpassword**



Note: Before starting the deployment of a new master domain manager or backup master domain manager on an already used database, be sure that no failed plan creation/extension has been performed. If a failed plan creation or extension has been performed, resolve the failure before attempting the new deployment or unlock the database by running the `planman unlock db` command.

5. If you are installing a backup master domain manager, it is crucial to use the same encryption keys as those on the master domain manager, to ensure it can correctly decrypt encrypted files, such as the Symphony file. To achieve this, perform the following steps:
 - a. Backup the files located in the `TWA_DATA_DIR\ssl\aes` folder on the backup master domain manager.
 - b. Copy the files from the `TWA_DATA_DIR\ssl\aes` folder on the master domain manager to the `TWA_DATA_DIR\ssl\aes` folder on the backup master domain manager.
6. To verify that the installation completed successfully, browse to the directory where you installed the master domain manager and type the following commands:

On UNIX operating systems

```
./twc_env.sh
```

On Windows operating systems

```
twc_env.cmd
```

```
optman ls
```

This command lists the HCL Workload Automation configurations settings and confirms that HCL Workload Automation installed correctly.

You can also optionally run `JnextPlan -for 0000` to extend by 0 hours and 0 minutes the production plan and add into the production plan (Symphony) the newly-created workstation, or wait for the FINAL job stream to complete, then run `composer list cpu=server_workstation_name` to ensure the agents have registered. You can also run a test job to ensure everything is working correctly.

Results

You have now successfully installed the master domain manager and backup master domain manager.

If you want to customize more installation parameters, see [FAQ - master domain manager and backup master domain manager customizations on page 107](#).

What to do next

You can proceed to [Installing the Dynamic Workload Console servers on page 110](#).

FAQ - master domain manager and backup master domain manager customizations

A list of questions and answers related to the customization of the master domain manager and backup master domain manager installation

When installing the master domain manager and backup master domain manager, you can perform a typical installation, as described in [Installing the master domain manager and backup master domain manager on page 100](#) or you can customize a number of parameters, as described in the following scenarios:

How do I customize general information for the master domain manager installation?

How to customize general information for the master domain manager installation.

How do I define the language of the messages?

To define the language in which messages are displayed, use the **-lang** parameter, as follows:

-lang lang_id

The language in which the `serverinst` messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used. If neither **-lang** nor LANG are used, the default codepage is set to SBCS. For a list of valid values for these variables, see the following table:

Table 4. Valid values for -lang and LANG parameter

Language	Value
Brazilian portuguese	pt_BR
Chinese (traditional and simplified)	zh_CN, zh_TW
English	en
French	fr
German	de
Italian	it
Japanese	ja
Korean	ko

Table 4. Valid values for -lang and LANG parameter
(continued)

Language	Value
Russian	ru
Spanish	es



Note: This is the language in which the installation log is recorded and not the language of the installed engine instance. `serverinst` installs all languages as default.

How do I modify the installation directory?

To modify the directory where the product is installed, use the **-inst_dir** parameter, as follows:

-inst_dir *installation_dir*

The directory of the HCL Workload Automation installation. This parameter is optional. The default value is calculated at installation time, based on the user performing the installation.

-work_dir *working_dir*

The temporary directory used by the program to deploy the installation process files. This parameter is optional. The default value is calculated at installation time, based on the user performing the installation.

I am confident that all my prerequisites are in order. How do I skip the prerequisites check?

To skip the prerequisites, use the **-skipcheckprereq** parameter, as follows:

-skipcheckprereq

If you set this parameter to `false`, HCL Workload Automation does not scan system prerequisites before starting the installation. This parameter is optional. The default value is `true`. For more information about the prerequisite check, see [Scanning system prerequisites for HCL Workload Automation on page 48](#).

How do I customize configuration information for the data source?

How to customize configuration information for the data source used by the master domain manager

How do I change the RDBMS type?

To use a different database than the default DB2, use the **-rdbms_type** parameter when typing the **serverinst** command, as follows:

--rdbms_type|-r *rdbms_type*

The database type. Supported databases are:

- **DB2**
- **ORACLE** This value applies to Oracle and Amazon RDS for Oracle

- **MSSQL** This value applies to MSSQL and MSSQL cloud-based databases.
- **POSTGRESQL**

This parameter is required and has no default value.

For more information about supported database versions, download the Supported Software report available at Product Requirements, then search for the **Databases** section.

I prefer not to use the default HCL Workload Automation database name (TWS). How do I change the database name?

To use a different database than the default DB2, use the **-dbname** parameter, as follows:

dbname *db_name*

Specify the name you want to use for the database. Note that this name must match the name specified in the `configureDb` command. For more information about the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

How can I specify a different database user?

To specify a different database name than the default value, use the **-dbuser** parameter, as follows:

dbuser *db_user*

Specify the name of the database user that accesses the HCL Workload Automation tables on the database server.

How can I specify a different database port?

To specify a different database port than the default value, use the **-dbport** parameter, as follows:

dbport *db_port*

Specify the port of the database server.

How do I customize configuration information for the master domain manager?

How to customize the configuration of the master domain manager

How can I customize the *data_dir* folder to maintain the previous behavior and store the data generated by HCL Workload Automation, such as logs, and configuration information together with the product binaries?

By default, at installation time product data and data generated by HCL Workload Automation, such as logs and configuration information are stored in the *data_dir* folder, separated from the product binaries.

If you want to revert to the previous behavior, where product data and product binaries were stored together, use the `--data_dir` argument to specify the HCL Workload Automation. For more information about the `--data_dir` argument, see [Server components installation - serverinst script on page 442](#).

You can also specify the `--data_dir` argument when installing the Dynamic Workload Console with the `dwcinst` command and the agents with the `twinst` command. For more information, see [Dynamic Workload Console installation - dwcinst script on page 456](#) and [Agent installation parameters - twinst script on page 119](#).

If you deploy the product components using Docker containers, the `<data_dir>` is set to the default directory name and location, and it cannot be modified.

How do I connect a new master domain manager to an existing Dynamic Workload Console?

Share certificates between a new master domain manager and an existing Dynamic Workload Console.

About this task

If you install a new master domain manager and you want it to connect to an existing Dynamic Workload Console, you need to import the master domain manager certificates into the Dynamic Workload Console keystore. For further information about how to import certificates by using Certman, see the topic about importing certificates from a master domain manager into the Dynamic Workload Console.

Installing the Dynamic Workload Console servers

Procedure for installing two Dynamic Workload Console servers on two separate nodes.

About this task



Procedure for installing the Dynamic Workload Console

About this task

In this scenario, the HCL Workload Automation administrator installs two Dynamic Workload Console instances on two separate workstations, sharing the same remote database. The HCL Workload Automation administrator performs the operations listed below on both workstations.

Certificates are now required when installing or upgrading HCL Workload Automation. You can no longer install nor upgrade HCL Workload Automation without securing your environment with certificates. The required certificates are:

- ca.crt
- tls.key
- tls.crt

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).



Note: If you use Db2 for z/OS with the Dynamic Workload Console version 10.2.4 or later, transfer the drivers in binary mode from the directory where you installed Db2 for z/OS to a directory of your choice. When you run the `configuredb` or `dwcinst` script, set the directory you chose in the **dbdriverspath** parameter.



Note: If you are installing the Dynamic Workload Console version 10.2.3 or later, the Federator is also automatically installed. This component enables you to monitor your objects through the Orchestration Monitor page of the Dynamic Workload Console. For detailed information about how to configure and use the Federator, see *Mirroring the z/OS current plan to enable the Orchestration Monitor* the section about mirroring the z/OS current plan to enable the Orchestration Monitor in the *Dynamic Workload Console User's Guide*.

If you are installing the on a z/OS operating system, see the topic about installing a Dynamic Workload Console server in *HCL Workload Scheduler for Z: Planning and Installation*.

The HCL Workload Automation administrator installs the Dynamic Workload Console. The following information is required:

Table 5. Required information

Command parameter	Required information	Provided in..
Database information		
--rdbmstype	database type	Creating and populating the database on page 59
--dbhostname	database hostname	
--dbport	database port	
--dbname	database name	
--dbuser	database user name	
--dbpassword	database password	
Security information		
--sslkeyfolder	name and path of the folder containing certificates	Installing the master domain manager and backup master domain manager on page 100
--sslpassword	password for the certificates	Installing the master domain manager and backup master domain manager on page 100
Open Liberty information		
--wlpdir	Open Liberty installation directory	Installing Open Liberty on page 56

You can run the **dwcinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

Default values are stored in the `dwcinst.properties` file, located in the root directory of the installation image.

If you need to modify any of the default values, edit the `dwcinst.properties` file, but do not modify the `dwcinst.template` file located in the same path.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

Before starting the Dynamic Workload Console installation, ensure the following steps have been completed:

1. [Installing Open Liberty on page 56](#) on the workstations where you plan to install the Dynamic Workload Console
2. [Creating and populating the database on page 59](#)
3. [Creating the HCL Workload Automation administrative user on page 99](#)
4. On UNIX™ operating systems, ensure that **umask** is set to **022**. To verify that **umask** is set to the correct value, from a command prompt, run the **umask** command. If the value is different from **022**, modify it by running the following command:

```
umask 022
```



Note:

- Ensure that the **inst_dir** parameter is different from the directory of the installation image and it does not contain any HCL Workload Automation instance.
- Recent JVMs do not fully support use of non-ASCII characters with the `-jar` and `-javaagent` commands. Use only ASCII characters in your installation directory names and paths.

To install the Dynamic Workload Console, perform the following steps:

1. Log in to the workstation where you plan to install the Dynamic Workload Console.
2. Download the installation images from [HCL Software](#).
3. Browse to the folder where the `dwcinst` command is located in `image_location/TWS/interp_name`.
4. Start the installation specifying a typical set of parameters:

On Windows operating systems

```
cscript dwcinst.vbs --acceptlicense yes --rdbmstype db_type
--user dwc_admin_user --password dwc_pwd --dbname db_name
--dbuser db_user --dbpassword db_pwd --dbhostname db_hostname
--dbport db_port --wlpdir Liberty_installation_dir\wlp
--sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
```

On UNIX operating systems

```
./dwcinst.sh --acceptlicense yes --rdbmstype db_type
--user dwc_admin_user --password dwc_pwd --dbname db_name
--dbuser db_user --dbpassword db_pwd --dbhostname db_hostname
--dbport db_port --wlpdir Liberty_installation_dir/wlp
--sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
```

where,

user *dwc_admin_user*

is the administrator of the Dynamic Workload Console. This user is added to the group of the Dynamic Workload Console administrators at installation time. You can use this account to log in to the Dynamic Workload Console and manage your environment.

password *dwc_pwd*

is the password of the Dynamic Workload Console user.

On Windows operating systems

Supported characters for the password are alphanumeric, dash (-), underscore (_), characters, and ()|?*~+.@!^

On UNIX operating systems

Supported characters for the password are any alphanumeric, dash (-), underscore (_), characters, and ()|?*~+.

Results

You have now successfully installed the Dynamic Workload Console.



Important: To ensure compatibility, the Dynamic Workload Console version installed must always be equal to or greater than the version of any engine it connects to.

For more information about all **dwcinst** parameters and default values, see [Dynamic Workload Console installation - dwcinst script on page 456](#).

What to do next

You can now proceed to [Installing agents on page 113](#).

Installing agents

How to install an HCL Workload Automation fault-tolerant agent or dynamic agent in your distributed or end-to-end network by using the twsinst script.

About this task



When you install a fault-tolerant agent, also the remote command-line client is installed. You can use the client to run `composer` and `conman` commands.

Use only the `twinst` script to install agents. If you are installing a dynamic agent, you can optionally add the Java™ run time which is needed to run job types with advanced options, and to configure a gateway to open communication with the dynamic workload broker.

When you install a dynamic or a fault-tolerant agent, also the following access methods, that extend the job scheduling capabilities of HCL Workload Automation to other software products, are installed:

PeopleSoft

To run and monitor PeopleSoft jobs from the HCL Workload Automation environment.

SAP R/3

To create, schedule, and control SAP jobs by using the job scheduling features of HCL Workload Automation.

z/OS

To define and schedule jobs that run in a z/OS environment with JES2, JES3, or HCL Workload Automation for Z

See the section about access methods in *Scheduling Job Integrations with HCL Workload Automation* for details about configuring and using the access methods.

During each step of the installation process, the `twinst` script creates files in the installation directory that you specified in the command. If you do not specify an installation directory in the `-inst_dir` option in the command, the script creates files in the following directories:

On Windows™ operating systems

```
%ProgramFiles%\HCL\TWA_TWS_USER
```

On UNIX™ operating systems

```
/opt/HCL/TWA_TWS_USER
```

Where `TWS_USER` is the user for which you are installing the HCL Workload Automation instance that you specify in the command.

The dynamic agent installation process automatically adds the workstation definition to the database and registers the workstation definition to the dynamic workload broker installed on the master domain manager or the dynamic domain manager that you specify during the installation process.

You can organize dynamic agents in pools to help organize your environment based on the availability of workstations and the requirements of the jobs to be run. Normally, when you create a pool, you add the dynamic agents to a workstation definition of type pool.

You can also register an agent with a pool by directly editing the `pools.properties` file located in `<TWS_home>/ITA/cpa/config`. See the topic about automatically registering agents to a pool in the *Planning and Installation Guide*.

To enable secure SSL communication for dynamic agents, you can choose one of the following methods:

- Download and deploy to dynamic agents the certificates already available on the master domain manager using the **wauser** and **wapassword** parameters when you run the `twinst` installation script. Ensure the certificates are available on the master domain manager in the `TWA_DATA_DIR/ssl/depot` path.
- Use the **sslkeyfolder** and **sslpassword** parameters when you run the `twinst` installation script. This applies to dynamic agents and fault-tolerant agents.

You only need to provide the path to the certificates and the password you want to define for the keystore and truststore. HCL Workload Automation automatically generates the keystore and truststore with the specified password and configures Open Liberty and your agents in SSL mode.

Enabling SSL during installation requires Java run time, which you can add at installation time using the **addjruntime** parameter, also available in the `twinst` installation script. For more information, see [Agent installation parameters - twinst script on page 119](#).

At installation time, you can optionally create a subfolder on the master domain manager to contain one or more `*.cert` files to be added to the server truststore as trusted CA using the **sslkeyfolder** parameter. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or Db2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**.

You can also use the **netmansslport** parameter when installing master domain manager, dynamic domain manager, and fault-tolerant agents to ensure **netman** communication between the server components and fault-tolerant agents takes place in SSL mode using the specified port number.

Agent installation procedure

Before you begin

1. Before you start to install, upgrade, or uninstall, verify that the user that runs the process has the following authorization requirements:

Windows™ operating system

If you set the Windows User Account Control (UAC), your login account must be a member of the Windows™ **Administrators** group or domain administrators with the rights **Act as Part of the Operating System**.

If you set the Windows User Account Control (UAC) on the workstation, you must run the installation as **administrator**.

UNIX™ and Linux™ operating systems

You can choose to install agents as the **root** user, or as a **user other than root**. The following considerations apply:

- If the installer is the **root user**, the **uname** parameter can be omitted if the *username* value is meant to be root, or can be set to a username value other than root.
 - If the installer is **different from the root user**, consider the following points:
 - The **uname** parameter can be omitted, but *username* is automatically set to the login name of the installer. If the installer specifies a **uname** with a different *username* value, an error message is returned.
 - As a consequence, the agent can run jobs uniquely with the user name of the installer.
 - The user must be enabled to login to the machine where the agent is going to be installed.
 - Event Management triggers on files work only if the selected files are accessible to the user that was used for the installation.
 - Future upgrades, modifications, and removal of the agent can be made exclusively with the same login used for installation. For dynamic agents, the login name used by the installer is stored in the read-only `InstallationLoginUser` parameter in the `JobManager.ini` configuration file on the agent.
 - When running **conman** and **composer** commands, it is mandatory to set the environment first, by using the `twc_env` script as described in [Setting the environment variables on page 206](#).
2. Ensure that you downloaded the agent elimages (for details, see `HWA_10.2.5_QuickStartGuide.zip` available from [HCL Software](#)).
 3. Ensure that you have enough temporary space before you start the installation process.

About this task

You can install an agent in a distributed or an end-to-end environment.

To install an HCL Workload Automation agent, perform the following steps:

On Windows™ operating systems:

1. Download the agent elimage. For more information, see `HWA_10.2.5_QuickStartGuide.zip` available from [HCL Software](#).
2. Log in as administrator on the workstation where you want to install the product.
3. From the `image_directory\TWS\operating_system` directory, run `twcinst` by using the following syntax:

```
cscript twcinst.vbs -new -uname username -password user_password -acceptlicense yes
```

For a description of the syntax parameters and a complete list of them, see [Agent installation parameters - twcinst script on page 119](#).



Note: `twinsinst` for Windows™ is a Visual Basic Script (VBS) that you can run in CScript and WScript mode.

The HCL Workload Automation user is automatically created. The software is installed by default in the HCL Workload Automation installation directory. The default value is `%ProgramFiles%\HCL\TWA`.

If you enabled the Security Warning, a dialog box is displayed during the installation. In this case answer Run to continue.

On UNIX™ and Linux™ operating systems:

1. Download the agent image. For more information about images, see [Downloading installation images on your workstation on page 232](#) or the `HWA_10.2.5_QuickStartGuide.zip` available from [HCL Software](#).
2. If you plan to login as **root** on the workstation where you will install the agent, create the HCL Workload Automation user. The software is installed by default in the user's home directory, referred to as `/installation_dir/TWS`.

User:

`TWS_user`

Home:

`/installation_dir/TWS` (for example: `/home/user1/TWS` where `user1` is the name of HCL Workload Automation user). Ensure this directory has **755** permission.

If you plan to log in as a **non-root user**, your login will become by default the only possible user of the agent. You do not need to create another HCL Workload Automation user, but make sure that you have a home directory (where the agent will be installed), and that it has **755** permission.



Important: If you use the `-su non-root username` command in the shell where you are about to run `twinsinst`, make sure that `$HOME` is set on your home directory as a non-root user (use `echo $HOME` to verify that the value returned corresponds to your home directory).

3. Log in on the workstation where you want to install the product.
4. From the `image_directory/TWS/operating_system` directory, run `twinsinst` by using the following syntax:

```
./twinsinst -new -uname username -acceptlicense yes
```

For a description of the syntax parameters, see [Agent installation parameters - twinsinst script on page 119](#)

If the installation fails, to understand the cause of the error see [Analyzing return codes for agent installation, upgrade, restore, and uninstallation on page 400](#).

After a successful installation, perform one of the following configuration tasks, depending on the type of agent you installed:

- [Configuring a fault-tolerant agent on page 140.](#)
- [Configuring a dynamic agent on page 216.](#)

On Windows™ operating systems:

Show command usage and version

```
cscript twsinst.vbs -u | -v
```

Install a new instance

```
cscript twsinst.vbs -new
  -acceptlicense yes|no
  -uname username
  [-domain user_domain]
  -password user_password

  [-agent dynamic|fta|both|zcentric]

  [-addruntime true|false]
  [-inst_dir install_dir]
  [-lang lang_id]
  [-skipcheckprereq]
  [-skip_usercheck]
  [-work_dir working_dir]

  [-agentid id]
  [-company company_name]
  [-master master_cpu_name]
  [-port port_number]
  [-netmansslport port_number]
  [-thiscpu workstation]
  [-encryptionpassword password]
  [-useencryption boolean]

  [-gateway local|remote|none]
  [-gwid gateway_id]
  [-gweifport gateway_eif_port]

  [-displayname agentname]
  [-hostname host_name]
  [-jimport port_number]
  [-jimportssl true|false]
  [-tdwbhostname host_name]
  [-tdwbport tdwbport_number]

  [-sslpassword ssl_password]
  [-sslkeysfolder ssl_folder]

  [-jwt true | false]
  [-wuser wuser -wapassword wapassword] | [-apikey apikey]
```

On UNIX™ and Linux™ operating systems

Show command usage and version

```
./twswinst -u | -v
```

Install a new instance

```
./twswinst -new
  -acceptlicense yes/no
  [-reset_perm]
  [-uname username]
  [-data_dir data_directory]
  [-agent dynamic|fta|both|zcentric]
  [-addjruntime true|false]
  [-inst_dir install_dir]
  [-lang lang_id]
  [-skipcheckprereq]
  [-skip_usercheck]
  [-work_dir working_dir]
  [-agentid id]
  [-company company_name]
  [-master master_cpu_name]
  [-port port_number]
  [-netmansslport port_number]
  [-thiscpu workstation]
  [-encryptionpassword password]
  [-useencryption boolean]
  [-gateway local|remote|none]
  [-gwid gateway_id]
  [-gweifport gateway_eif_port]
  [-displayname agentname]
  [-hostname host_name]
  [-jimport port_number]
  [-jimportssl true|false]
  [-tdwbhostname host_name]
  [-tdwbport tdwbport_number]
  [-create_link]
  [-sslpassword ssl_password]
  [-sslkeysfolder ssl_folder]
  [-jwt true | false]
  [-wauser wauser -wapassword wapassword] | [-apikey apikey]
```

Agent installation parameters - twswinst script

Agent installation parameters that can be passed to the twswinst script.

About this task

This section lists and describes the parameters that are used when running a twswinst script to install , , or (also known as agent with z-centric capabilities).



Note: To install a on HCL Universal Orchestrator, see [Installing and connecting HCL Workload Automation dynamic agents at version 10.2.3 or later](#).

To find some sample agent installation scenarios, see [Example installation commands on page 132](#) and [Dynamic agent gateway installation examples on page 135](#).

To manage authentication and certificates effectively, see [Table 7: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 130](#) for a comprehensive list of supported combinations for the following parameters:

- [-apikey on page 121](#)
- [-jwt true | false on page 124](#)
- [-sslkeysfolder path on page 126](#)
- [-sslpassword password on page 127](#)
- [-tdwbhostname host_name on page 127](#)
- [-tdwbport tdwbport_number on page 127](#)
- [-wapassword wauser_password on page 129](#)
- [-wauser wauser_name on page 129](#)

-acceptlicense *yes/no*

Specifies whether to accept the License Agreement.

-addjruntime *true/false*

Adds the Java™ run time to run job types with advanced options, both those types that are supplied with the product and the additional types that are implemented through the custom plug-ins. Valid values are **true** and **false**. The default for a fresh installation is **true**. Set this parameter to `true` if you use the **sslkeysfolder** and **sslpassword** parameters to define custom certificates in PEM format.

If you decided not to install Java™ run time at installation time, you can still add this feature later as it is described in [Adding a feature on page 225](#).

-agent *dynamic/fta/both/zcentric*

Specifies the type of agent to install. Valid values are:

dynamic

Installs the dynamic agent. **Requires** the **-tdwbhostname** *host_name* and the **-tdwbport** *tdwbport_number* parameters.

fta

Installs the fault-tolerant agent.

both

Installs the dynamic agent that is used with the **-tdwbhostname** *host_name* and the **-tdwbport** *tdwbport_number* parameters, and a fault-tolerant agent.

zcentric

Installs the (also known as agent with z-centric capabilities).

The default is **dynamic**.

-agentid *agent_id*

Specifies a unique identifier for the agent. If not provided, an alphanumeric ID is automatically generated

```
893164748CCA4FC6820F12685AECBB07
```

To reuse an agent ID for reinstallations, specify the same `agent_id`. Ensure the value is exactly 32 characters long; otherwise, an error occurs.

If you set the **jwt** parameter to `true`, the **agentid** parameter is ignored if provided, because the agent ID is retrieved from the together with the JWT. See [-jwt true | false on page 124](#).

-apikey

Specifies the API key for authentication with the . This key enables downloading certificates or JWT for communication between and . A random password in base64 encoding is automatically created for generating stash files. The password stored in the `tls.sth` file. If needed, you can decrypt this password using any base64 decoder.

Obtain the string to be provided with this parameter from the before running the command. For more information, see the section about authenticating the command line client using API Keys in .

This parameter is **mutually exclusive** with:

- [-wauser wauser_name on page 129](#)
- [-wapassword wauser_password on page 129](#)
- [-sslkeysfolder path on page 126](#)
- [-sslpassword password on page 127](#)

and it is **required** with:

- [-tdwbhostname host_name on page 127](#)
- [-tdwbport tdwbport_number on page 127](#)

For a comprehensive list of supported combinations for this parameter and related ones, see [Table 7: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 130](#).

-company *company_name*

The name of the company. The company name cannot contain blank characters. The name is shown in program headers and reports. If not specified, the default name is COMPANY.

-create_link

UNIX™ systems only. Create the **symlink** between `/usr/bin/at` and `install_dir/TWS/bin/at`. For more information, see [Table 2: Symbolic link options on page 34](#).

-data_dir path

This argument applies to UNIX operating systems only. Specify a path for product data, such as log and configuration files, if you want to install the product binaries separated from the product data. This argument is optional. The default value is `INSTALL_DIR/TWSDATA`.

-displayname display_name

The name to assign to the agent. The name cannot start with a number. The default is based on the host name of this computer.

If the host name starts with a number, the **-displayname** parameter must be specified.

-domain user_domain

Windows™ systems only. The domain name of the user. The default is the name of the workstation on which you are installing the product. Ensure you use `USERDOMAIN` instead of `USERDNSDOMAIN`.

-enablefips true/false

Specify whether you want to enable FIPS. The default value is `false`. This parameter is optional.

-encryptionpassword default**-gateway local/remote/none**

Specifies whether to configure a gateway to communicate with the or not, and how it is configured. Specify `local` if the gateway is local to the workstation. Specify `remote` if the communicates through a gateway that is installed on a different workstation from the being installed. The default value is `none`, which means no gateway is configured. For information about installing with a local and remote gateway, see [Example installation commands on page 132](#).

-gweifport gateway_eif_port

Specifies the Job Manager Event Integration Facility (EIF) port number. The default value is **31132**. The valid range is 1 to 65535.

-gwid gateway_id

The unique identifier for the gateway. This parameter is required when you specify **-gateway local** and must be unique across all agents. The default gateway identifier that is assigned is **GW1**. The gateway identifier must start with either an alphabetic character or an underscore character (`_`), and it can contain only the following types of characters: alphabetic, numeric, underscores (`_`), hyphens (`-`), and periods (`.`).

Gateways can also work in parallel to mutually take over in routing communications to the agents connected to them. To enable gateways to work in parallel, all gateways must have the same `gateway_id` assigned. This information is stored in the `JobManagerGW.ini` file, by setting the **JobManagerGWURIs** property.

-hostname host_name

The fully qualified hostname or IP address on which the agent is contacted by the . The default is the hostname of this computer. If the hostname is a localhost, the hostname parameter must be specified.

-inst_dir *installation_dir*

The directory of the installation.

On Windows™ operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. Specify an absolute path. If you do not manually specify a path, the path is set to %ProgramFiles%\HCL\TWA_TWS_USER, where *TWS_USER* is the user for which you are installing the that you specify in the **-uname** parameter. If you use the Local System Account and therefore do not specify the **-uname** parameter, the path is set to %ProgramFiles%\HCL\TWA_WaLocalSystemAccount.

On UNIX™ and Linux™ operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. Specify an absolute path. If you do not manually specify a path, the path is set to:

- `/opt/HCL/TWA_TWS_USER`, if you logged in as the **root** user to install the agent. *TWS_USER* is the user that you specify in the **-uname** option and for which you are installing the agent (can omit if *TWS_USER* is **root**). The user that you specify in the **-uname username** parameter must have read and run privileges for the *installation_dir* installation path; otherwise the installation fails.
- `home_dir/TWA`, if you logged in with a login **other than root**. Ensure that the directory permission is set to **755** for *home_dir*, the home directory for your login, and that you are the *home_dir* owner.

-jimport *port_number*

The JobManager port number used by the to connect to the . The default value is **31114**. The valid range is from 1 to 65535.

-jimportssl *true/false*

The JobManager port used by the to connect to the . The port value is the value of the *ssl_port* parameter in the *ita.ini* file if **-jimportssl** is set to *true*. If set to *false*, it corresponds to the value of the *tcp_port* parameter in the *ita.ini* file. The *ita.ini* file is located in *ITA\cpa\ita* on Windows™ systems and *ITA/cpa/ita* on UNIX™, Linux™, and systems.

Set the value to "true" if **- gateway** is set to *local*.

For communication using SSL or HTTPS

Set **jimportssl = true**. To communicate with the , it is recommended that you set the value to *true*. In this case, the port specified in **jimport** communicates in HTTPS.

For communication without using SSL or through HTTP

Set **jimportssl = false**. In this case the port specified in **jimport** communicates in HTTP.

-jwt *true* / *false*

Specify `true` to use a JSON Web Token (JWT) for authentication with the . Specify `false` to authenticate with the using certificates instead. The default value is `true`.

When set to `true`, this parameter is **mutually exclusive** with the following parameters which are used to generate custom certificates:

- [-sslkeysfolder path on page 126](#)
- [-sslpassword password on page 127](#)

If you set this parameter to `true`, note the following:

- the [-agentid agent_id on page 121](#), if provided, will be ignored because the agent ID is retrieved from the along with the JWT.
- The following parameters are **required** for downloading the JWT:
 - [-wouser wouser_name on page 129](#) or [-apikey on page 121](#).
 - [-wapassword wouser_password on page 129](#) or [-apikey on page 121](#).
 - [-tdwbhostname host_name on page 127](#). This parameter is always required when **jwt** is set to `true`, regardless of whether you use the **wouser** and **wapassword** or the **apikey** parameters.
 - [-tdwbport tdwbport_number on page 127](#). This parameter is always required when **jwt** is set to `true`, regardless of whether you use the **wouser** and **wapassword** or the **apikey** parameters.

For examples of installations with JWT, see [Example installation commands on page 132](#).

For a comprehensive list of supported combinations for this parameter and related ones, see [Table 7: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 130](#).

-lang *lang_id*

The language in which the twsinst messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used. If neither **-lang** nor LANG are used, the default codepage is set to SBCS. For a list of valid values for these variables, see the following table:

Table 6. Valid values for -lang and LANG parameter

Language	Value
Brazilian portuguese	pt_BR
Chinese (traditional and simplified)	zh_CN, zh_TW
English	en
French	fr

Table 6. Valid values for -lang and LANG parameter
(continued)

Language	Value
German	de
Italian	it
Japanese	ja
Korean	ko
Russian	ru
Spanish	es



Note: This is the language in which the installation log is recorded and not the language of the installed engine instance. twsinst installs all languages as default.

-master workstation

The workstation name of the master domain manager. This name cannot exceed 16 characters, cannot contain spaces, and cannot be the same as the workstation name that you entered in the **thiscpu** parameter. If not specified, the default value is **MASTER**.

-netmansslport SSL_port_number

-new

A fresh installation of the agent. Installs an agent and all supported language packs.

-password user_password

Windows™ systems only. The password of the user for which you are installing . The password can include alphanumeric, dash (-), and underscore (_) characters, and the following symbols: (!)?=^*/~ [] \$'+;.:@. The **-password** parameter is used for fresh installations only, it is not required for fix packs or upgrades. You can optionally encrypt the password using the secure script. For more information, see [Optional password encryption - secure script on page 427](#).

-port port_number

The TCP/IP port number used by the Netman process to listen for communication from the master. The default value is **31111**. The valid range is from 1 to 65535. This port number is registered in the `localopts` file. For each installation you must specify a different number.

-reset_perm

UNIX™ and systems only. Reset the permission of the libraries in the `/usr/hcl` directory.

-restore

Run this command from the folder to where you copied the elmage (a folder other than the home directory of *TWS_USER*, where *TWS_USER* is the user that installed the instance), and not from the installation path, to restore the version in the elmage.

-skip_usercheck

Enable this option if the authentication process within your organization is not standard, thereby disabling the default authentication option.

On Windows™ systems, if you specify this parameter, the program does not create the user you specified in the **-uname** *username* parameter and you must create the user manually before running the script. However, if you use Local System Account, you do not need to specify any user.

On UNIX™ and Linux™ systems if you specify this parameter, the program skips the check of the user in the */etc/passwd* file or the check you perform using the *su* command.

-skipcheckprereq

If you specify this parameter, does not scan system prerequisites before installing the agent. For more information on the prerequisite check, see [Scanning system prerequisites for HCL Workload Automation on page 48](#).

-sslkeyfolder path

The name and path of the folder on the agent containing PEM certificates. The installation program automatically generates the keystore and truststore files using the password you specify with the **sslpasword** parameter, which is **required** when using **sslkeyfolder**.

The folder must contain the following files and folders:

ca.crt

The Certificate Authority (CA) public certificate.

tls.key

The private key for the instance to be installed.

tls.crt

The public part of the previous key.

tls.sth

The file storing your encoded password in Base64 encoding.

You can optionally create a subfolder to contain one or more *.*crt* files to be added to the server truststore as trusted CA. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or DB2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**. If you are connecting a using custom certificates to a also using custom certificates, the only required file is *ca.crt*.

Before you start the installation, ensure the required files and folders are available on the agent.

The **sslkeysfolder** and **sslpassword** parameters are **mutually exclusive** with the **wauser**, **wapassword**, **apikey**, and **jwt true** parameters, which are used to download and deploy the certificates or JWT already available on the .

For a comprehensive list of supported combinations for this parameter and related ones, see [Table 7: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 130](#).

-sslpassword *password*

Specify the password for the certificates in PEM format automatically generated by the installation program. It requires the **sslkeysfolder** parameter.

If you use this parameter, ensure that the **addruntime** parameter is set to true, because Java™ run time is required for defining custom certificates.

The **sslkeysfolder** and **sslpassword** parameters are **mutually exclusive** with the **wauser**, **wapassword**, **apikey**, and **jwt true** parameters, which are used to download and deploy the certificates or JWT already available on the .

For a comprehensive list of supported combinations for this parameter and related ones, see [Table 7: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 130](#)

-tdwbhostname *host_name*

The fully qualified host name or IP address of the the agent is registering to. This parameter **requires** the **-tdwbport** parameter. It **is required** if you use the **wauser** and **wapassword** or the **apikey** parameters. This parameter is not supported on (also known as agent with z-centric capabilities).

If you set the **-gateway** parameter to `remote`, this is the host name of the hosting the gateway and to which the agent you are installing will connect. This information is stored in the `JobManager.ini` file. For information about installing with a local and remote gateway, see [Example installation commands on page 132](#).

See also [-jwt true | false on page 124](#).

For a comprehensive list of supported combinations for this parameter and related ones, see [Table 7: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 130](#).

-tdwbport *tdwbport_number*

The HTTPS transport port number of the the agent is registering to. It must match the port you specified with **httpsport** parameter when installing the . It is **required** if you use the **wauser** and **wapassword** or the **apikey** parameters and **requires** the **-tdwbhostname** parameter. This parameter is not supported on (also known as agent with z-centric capabilities).

The valid range is from 0 to 65535. If you specify 0 you cannot run workload dynamically. Do not specify 0 if the `-agent` value is `dynamic` or `both`. The default is 0 for an upgrade, which means that this connection is not configured, otherwise, specify 31116 for a fresh installation.

If you set the `-gateway` parameter to `remote`, this is the HTTP or HTTPS port number of the host hosting the gateway and to which the agent you are installing will connect. You have specified this port with the `import` parameter when installing the agent hosting the gateway. For information about installing with a local and remote gateway, see [Example installation commands on page 132](#).

If you are performing a fresh installation, the value to use is 31114. This information is stored in the `JobManager.ini` file.

For a comprehensive list of supported combinations for this parameter and related ones, see [Table 7: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 130](#).

-thiscpu workstation

The name of the workstation of this installation. The name cannot exceed 16 characters, cannot start with a number, cannot contain spaces, and cannot be the same as the workstation name of the master domain manager. This name is registered in the `localopts` file. If not specified, the default value is the host name of the workstation.

If the host name starts with a number, `-thiscpu` parameter must be specified.

-u

Displays command usage information and exits.

-uname username

The name of the user for which the agent is being installed. This user owns the instance and by default, jobs are run with its name. This user name is not to be confused with the user performing the installation, unless you use a **user other than root**. The user name cannot contain periods (.).

On UNIX™ and Linux™ systems, for a new installation, this user account must be created manually before running the installation and must be enabled to login to the machine where the agent is going to be installed. Create a user with a home directory. `is` is installed by default under the home directory of the specified user.

On Windows operating systems, you can install `and` using the Local System Account by omitting the `uname` and `password` parameters.

-useencryption true | false

-v

Displays the command version and exits.

-wapassword *wauser_password*

One of the following passwords, defined on the :

- The password of the user for which you have installed the the agent is connecting to.
- The password of the user with the DISPLAY permission on the FILE named AGENT_CERTIFICATE. This permission allows the user to download certificates or JWT. For more information about this scenario, see [Downloading certificates or JWT using a different user on page 487](#).

See also [-jwt true | false on page 124](#).

For further information about how to automatically download and deploy certificates in PEM format from the master domain manager to dynamic agents and fault-tolerant agents, see [Certificates download to dynamic agents and fault-tolerant agents - AgentCertificateDownloader script on page 482](#).

This parameter always requires the [tdwbport on page 127](#) and [-tdwbhostname host_name on page 127](#) parameters.

For a comprehensive list of supported combinations for this parameter and related ones, see [Table 7: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 130](#).

-wauser *wauser_name*

One of the following users, defined on the :

- The user for which you have installed the the agent is connecting to.
- The user with the DISPLAY permission on the FILE named AGENT_CERTIFICATE. This permission allows the user to download certificates or JWT. For more information about this scenario, see [Downloading certificates or JWT using a different user on page 487](#).

Always specify the user defined on the , also if you are installing a and want it to register to a . This is because the simply forwards data to and from the .

By providing the **wauser** and **wapassword** parameters or the **apikey** parameter, you enable to download and install either the certificates or the JWT already available on the :

- To download certificates, set the **jwt** parameter to `false`
- To download JWT, set the **jwt** parameter to `true`. For more information, see [-jwt true | false on page 124](#).

Key details about this parameter:

- It is **mutually exclusive** with the [-apikey on page 121](#) parameter, which provides authentication using an API Key and the [-sslkeysfolder path on page 126](#) and [-sslpassword password on page 127](#) parameters.
- It **always requires** the [tdwbport on page 127](#) and [-tdwbhostname host_name on page 127](#) parameters.
- It is **not supported** on the (also known as the agent with z-centric capabilities). To generate certificates for the , use the **sslkeysfolder** and **sslpassword** parameters.

For further information about how to automatically download and deploy certificates in PEM format from the master domain manager to dynamic agents and fault-tolerant agents, see [Certificates download to dynamic agents and fault-tolerant agents - AgentCertificateDownloader script on page 482](#).

For a comprehensive list of supported combinations for this parameter and related ones, see [Table 7: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 130](#).

-work_dir working_dir

The temporary directory used by the program to deploy the installation process files.

On Windows™ operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. If you do not manually specify a path, the path is set to %temp%\TWA\twsversion_number, where %temp% is the temporary directory of the operating system.

On UNIX™ and Linux™ operating systems:

The path cannot contain blanks. If you do not manually specify a path, the path is set to /tmp/TWA/twsversion_number.

This parameter can also function as a backup directory during product upgrade with path `WORKING_DIR/backup` if you do not set the **-skipbackup** parameter to **true**.

Table 7. Supported combinations and mutual exclusions for authentication and certificate-related parameters

Parameter	Compatibility	Mutual exclusion	Required with
-apikey	Used to authenticate with the master domain manager and download certificates or JWT.	Cannot be used with: <ul style="list-style-type: none"> • -wauser • -wapassword • -sslkeysfolder • -sslpassword 	<ul style="list-style-type: none"> • -tdwbhostname • -tdwbport
-jwt false	Uses certificates for authentication.	No direct exclusion	EITHER

Table 7. Supported combinations and mutual exclusions for authentication and certificate-related parameters (continued)

Parameter	Compatibility	Mutual exclusion	Required with
			<ul style="list-style-type: none"> • -apikey • -tdwbhostname • -tdwbport OR <ul style="list-style-type: none"> • -wuser • -wapassword • -tdwbhostname • -tdwbport OR <ul style="list-style-type: none"> • -sslkeyfolder • -sslpassword
-jwt true	Uses JWT for authentication.	Cannot be used with: <ul style="list-style-type: none"> • -sslkeyfolder • -sslpassword 	EITHER <ul style="list-style-type: none"> • -apikey • -tdwbhostname • -tdwbport OR <ul style="list-style-type: none"> • -wuser • -wapassword • -tdwbhostname • -tdwbport
-sslkeyfolder	Specifies a folder containing PEM certificates.	Cannot be used with <ul style="list-style-type: none"> • -apikey • -wuser • -wapassword • -jwt true 	<ul style="list-style-type: none"> • -sslpassword
-sslpassword	Password for PEM certificates.	Cannot be used with <ul style="list-style-type: none"> • -apikey • -wuser 	<ul style="list-style-type: none"> • -sslkeyfolder

Table 7. Supported combinations and mutual exclusions for authentication and certificate-related parameters (continued)

Parameter	Compatibility	Mutual exclusion	Required with
		<ul style="list-style-type: none"> • -wapassword • -jwt true 	
-tdwbhost name	Hostname/IP of the Dynamic Workload Broker (DWB).	No direct exclusion	EITHER <ul style="list-style-type: none"> • -apikey • -tdwbport OR <ul style="list-style-type: none"> • -wauser • -wapassword • -tdwbport
-tdwbport	HTTPS port of the Dynamic Workload Broker (DWB).	No direct exclusion	EITHER <ul style="list-style-type: none"> • -tdwbhostname • -apikey OR <ul style="list-style-type: none"> • -tdwbhostname • -wauser • -wapassword
-wauser	User for authentication and downloading certificates/JWT	Cannot be used with <ul style="list-style-type: none"> • -apikey • -sslkeysfolder • -sslpassword 	<ul style="list-style-type: none"> • -wapassword • -tdwbhostname • -tdwbport
-wapassw ord	.Password of the -wauser user.	Cannot be used with <ul style="list-style-type: none"> • -apikey • -sslkeysfolder • -sslpassword 	<ul style="list-style-type: none"> • -wauser • -tdwbhostname • -tdwbport

Example installation commands

About this task

Consider the following examples to understand the use and capabilities of the **twinst** command. You can optionally encrypt the password using the secure script. For more information, see [Optional password encryption - secure script on page 427](#).

1. The following example shows the syntax used when using the **twinst** script to install a new instance of a dynamic agent and adding the Java™ run time for running job types with advanced options.

On Windows™ operating systems:

```
cscript twinst.vbs -new
  -uname TWSuser1 -password user_password -acceptlicense yes
  -addjruntime true -agent dynamic -displayname thishostcomputername
  -hostname thishostname.mycompany.com -wauser wauser -wapassword wapassword
  -inst_dir "c:\Program Files\HCL\TWA_TWSuser1"
  -jimport 31114 -tdwbport 31116 -tdwbhostname mainbroker.mycompany.com
```

On UNIX and Linux™ operating systems:

```
./twinst -new
  -uname TWSuser1 -acceptlicense yes -addjruntime true
  -agent dynamic -displayname thishostcomputername
  -hostname thishostname.mycompany.com -wauser wauser -wapassword wapassword
  -inst_dir "/opt/HCL/TWA_TWSuser1"
  -jimport 31114 -reset_perm -skipcheckprereq -tdwbport 31116
  -tdwbhostname mainbroker.mycompany.com
```

2. The following example shows the syntax used when running the **twinst** script to install a new instance of both a fault-tolerant and a dynamic agent, and adding the Java™ run time for running job types with advanced options. Ensure you copy the certificates on the agent before you start the installation. The path to the certificates is specified with the **sslkeysfolder** parameter. The **sslpassword** parameter specifies the password to access the certificates. In this case, the **jwt** parameter must be set to `false`:

On Windows™ operating systems:

```
cscript twinst.vbs -new
  -uname TWSuser1 -password user_password -acceptlicense yes
  -addjruntime true -agent both -displayname thishostcomputername
  -hostname thishostname.mycompany.com -sslkeysfolder /MyCertsFolder -jwt=false
  -sslpassword fer1smx24569ijDCS86?! -inst_dir "c:\Program Files\HCL\TWA_TWSuser1"
  -jimport 31114 -master TWSmdm -tdwbport 31116
  -tdwbhostname mainbroker.mycompany.com
  -thiscpu mainworkstation
```

On UNIX™ and Linux™ operating systems:

```
./twinst -new
  -uname TWSuser1 -acceptlicense yes -addjruntime true
  -agent both -create_link -displayname thishostcomputername
  -hostname thishostname.mycompany.com -inst_dir "/opt/HCL/TWA_TWSuser1"
  -sslkeysfolder /MyCertsFolder -sslpassword fer1smx24569ijDCS86?!
  -jimport 31114 -master TWSmdm -reset_perm -skipcheckprereq
  -tdwbport 31116 -tdwbhostname mainbroker.mycompany.com
  -thiscpu fta101
```

3. The following example shows the syntax used when using the **twsinst** script to install a new instance of a dynamic agent, adding the Java™ run time for running job types with advanced options, and to install a gateway on the same workstation as the agent to enable communication with the master domain manager.

On Windows™ operating systems:

```
cscript twsinst.vbs -new
    -uname TWSuser1 -password user_password -acceptlicense yes
    -addjruntime true -agent dynamic -displayname thishostcomputername
    -gateway local -gwid gateway_id
    -hostname thishostname.mycompany.com
    -inst_dir "c:\Program Files\HCL\TWA_TWSuser1"
    -wuser MDMAAdmin -wapassword 547832gtrOLK8542Mnfdw!
    -jimport 31114 -jimportssl true -master TWSmdm -skipcheckprereq
    -tdwbport 31116 -tdwbhostname mainbroker.mycompany.com
    -thiscpu mainworkstation
```

On UNIX™ and Linux™ operating systems:

```
./twsinst -new
    -uname TWSuser1 -acceptlicense yes -addjruntime true -agent both
    -displayname thishostcomputername -create_link -gateway local
    -gwid gateway_id
    -hostname thishostname.mycompany.com -inst_dir "/opt/HCL/TWA_TWSuser1"
    -wuser MDMAAdmin -wapassword 547832gtrOLK8542Mnfdw!
    -jimport 31114 -jimportssl true -master TWSmdm -reset_perm -skipcheckprereq
    -tdwbport 31116 -tdwbhostname mainbroker.mycompany.com
    -thiscpu fta101
```

4. In the following example, you install a new agent with both dynamic agent and fault-tolerant agent capabilities.

By setting **jwt** to **true**, you install the agent and authenticate with the master domain manager using JWT. The **jwt** parameter requires the **tdwbhostname** and **tdwbport** parameters for connecting to the dynamic domain manager and the **wuser** and **wapassword** parameters, which provide the credentials to be used when logging in to the master domain manager. The credentials are required when you first download the JWT from the master domain manager:

```
twsinst -new -acceptlicense yes -agent both -uname TWSuser1 -tdwbhostname Saturn
    -tdwbport 37116 -master Jupiter -jwt true -wuser MDMAAdmin -wapassword 125784gtrOLK8542Mnfdw!
```

5. This example is a variation from the previous example. Instead of using the **wuser** and **wapassword** parameters when logging in to the master domain manager, you authenticate using the API Key you have previously retrieved from the Dynamic Workload Console. For this purpose, use the **apikey** parameter:

```
twsinst -new -acceptlicense yes -agent both -uname TWSuser1 -tdwbhostname Saturn
    -tdwbport 37116 -master Jupiter -jwt true -apikey eyJraWQiOiJha2li...KINijWmdC-fY
```

6. In the following example, you install a dynamic agent and authenticate with the master domain manager named **Saturn** using JWT. The agent hosts a **local** gateway with gateway ID **GWID1**:

```
twsinst -new -acceptlicense yes -agent dynamic -uname TWSuser1 -hostname Titan -displayname Titan
    -tdwbhostname Saturn -tdwbport 37116 -jwt true -wuser MDMAAdmin
    -wapassword 125784gtrOLK8542Mnfdw! -gateway local -gwid GWID1 -jimport 42427
```

7. In the following example, you install an agent and authenticate with the master domain manager using JWT. The agent connects to the **remote** gateway installed on the agent in the previous example. In this case, the

tdwbhostname parameter must be set to the host name of the dynamic agent (*Titan*) hosting the gateway and to which the agent you are installing will connect. In the same way, the **tdwbport** parameter must match the port number of the dynamic agent hosting the gateway. You have set this port using the **jimport** parameter when installing the dynamic agent hosting the gateway.

```
twshint -new -acceptlicense yes -agent both -uname TWSuser1 -displayname Tetis
-tdwbhostname Titan -tdwbport 42427 -jwt true -wauser MDMAAdmin
-wapassword 125784gtr0LK8542Mnfdw! -gateway remote -jimport 54548
```

8. In the following example, you install the agent using local certificates for authentication. Ensure you copy the certificates on the agent before you start the installation. The path to the certificates is specified with the **sslkeyfolder** parameter. The **sslpassword** parameter specifies the password to access the certificates. In this case, the **jwt** parameter must be set to *false*:

```
twshint -new -acceptlicense yes -agent both -uname TWSuser1 -tdwbhostname Saturn
-tdwbport 37116 -master Jupiter -jwt false -sslkeyfolder /MyCertsFolder
-sslpassword fer1smx8854ijDSW65?!
```

9. In the following example, you install the agent and download to the agent the certificates from the master domain manager. Ensure the certificates are available on the master domain manager in one of the following paths:

On Windows operating systems

```
installation_directory\TWS\ssl\depot
```

On UNIX operating systems

```
TWA_DATA_DIR/ssl/depot
```

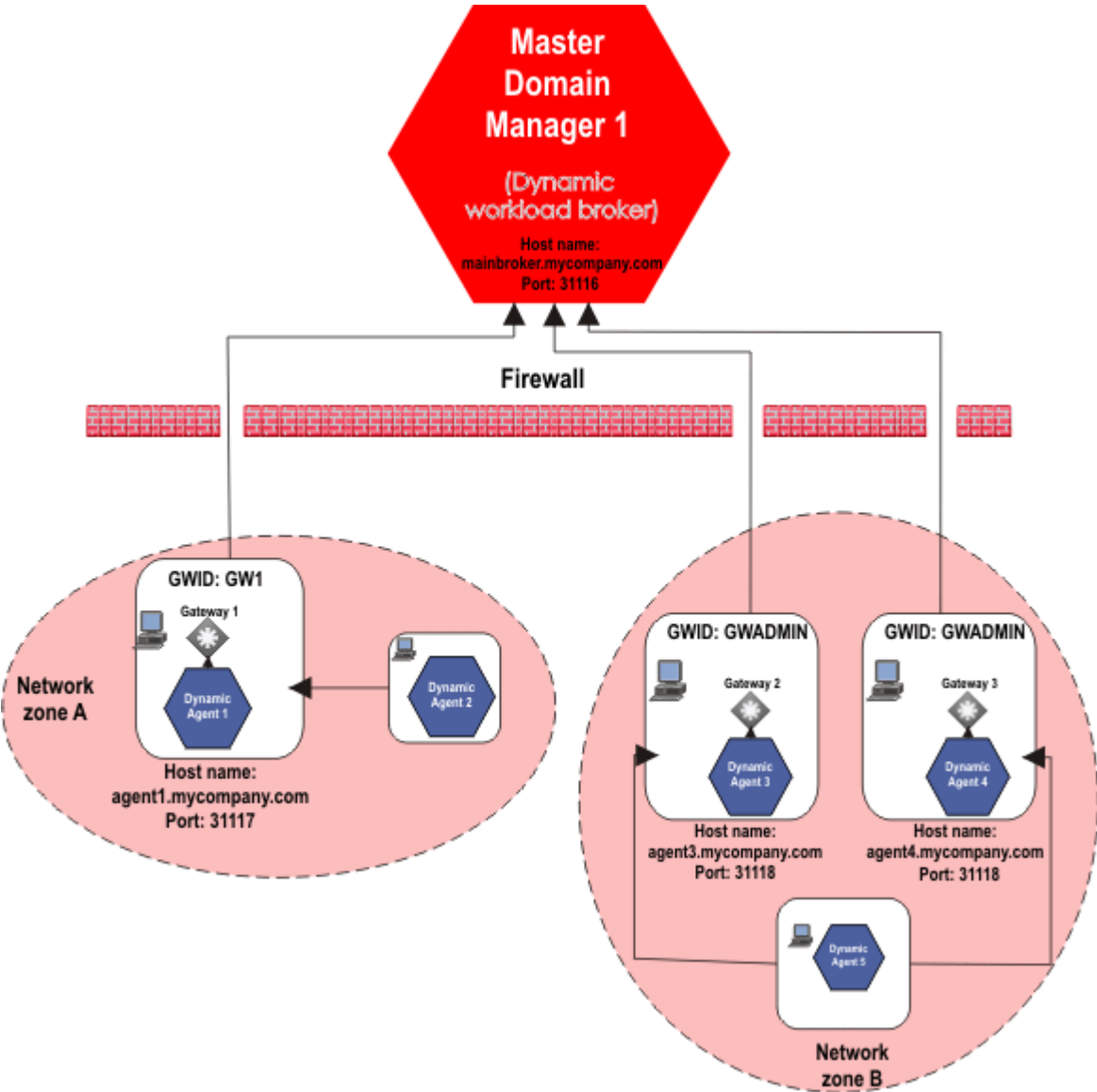
To download the certificates, specify the **wauser** and **wapassword** parameters to connect to the master domain manager. In this case, the **jwt** parameter must be set to *false*

```
twshint -new -acceptlicense yes -agent both -uname TWSuser1 -tdwbhostname Saturn -tdwbport 37116
-master Jupiter -jimport 1234 -port 1235 -netmansslport 1236 -jwt false -wauser MDMAAdmin
-wapassword 125784gtr0LK8542Mnfdw!
```

Dynamic agent gateway installation examples

Example installations for configuring a local or remote gateway with dynamic agent workstations in the same or different network zones.

The following examples address two installation scenarios and indicate the parameters to use with the **twshint** script to install the dynamic agents to support the scenarios. The following figure depicts the two scenario environments:



Scenario 1: Same network zone

The workstations where you install the agents can communicate with each other (Dynamic Agent 1 and Dynamic Agent 2) and are located in the same network zone, but only one agent workstation (Dynamic agent 1) can connect to the dynamic workload broker.

Table 8. Installation syntax for agent installation with agents in the same network zone

Dynamic Agent workstation	Installation syntax
Dynamic Agent 1	<pre>twinst -new -uname <user_name> -password <user_password> -acceptlicense yes -agent dynamic -gateway local</pre>

Table 8. Installation syntax for agent installation with agents in the same network zone (continued)

Dynamic Agent workstation	Installation syntax
	<pre>-gwid GW1 -jmport 31117 -tdwbport 31116 -tdwbhostname mainbroker.mycompany.com -wauser wauser -wapassword password</pre>
Dynamic Agent 2	<pre>twsinst -new -uname <user_name> -password user_password> -acceptlicense yes -agent dynamic -gateway remote -tdwbport 31117 -tdwbhostname agent1.mycompany.com</pre>

where,

Dynamic Agent 1

-gateway local

Dynamic Agent 1 communicates with the dynamic workload broker through its local gateway.

-gwid GW1

The gateway ID is the name that identifies the gateway site on Dynamic Agent 1. The default name is GW1.

-tdwbport 31116

The port number of the dynamic workload broker.

-tdwbhostname mainbroker.mycompany.com

The fully qualified host name of the dynamic workload broker.

Dynamic Agent 2

-gateway remote

Indicates that Dynamic Agent 2 can connect to the internet through a gateway installed on a different agent, Dynamic Agent 1.

-tdwbport 31117

The port number of the dynamic agent workstation where the gateway resides. In this example, the port number of Dynamic Agent 1 is 31117.

-tdwbhostname agent1.mycompany.com

The fully qualified host name of the dynamic agent workstation where the gateway resides and to which the agent connects.

Scenario 2: Different network zones

The workstations where you install the agents cannot communicate with each other and are in different network zones (Network zone A and Network zone B), however, one agent workstation in each network zone can successfully connect to the dynamic workload broker. In Network zone B, two parallel gateways are configured.

Table 9. Installation syntax for agent installation with agents in different network zones

Dynamic Agent workstation	Installation syntax
Dynamic Agent 3	<pre>twsinst -new -uname <user_name> -password user_password -acceptlicense yes -agent dynamic -gateway local -gwid GWADMIN -jimport 31118 -tdwbport 31116 -tdwbhostname mainbroker.mycompany.com</pre>
Dynamic Agent 4	<pre>twsinst -new -uname <user_name> -password user_password -acceptlicense yes -agent dynamic -gateway local -gwid GWADMIN -jimport 31118 -tdwbport 31116 -tdwbhostname mainbroker.mycompany.com</pre>
Dynamic Agent 5	<pre>twsinst -new -uname <user_name> -password user_password -acceptlicense yes -agent dynamic -gateway remote -tdwbport 31118 -tdwbhostname agent4.mycompany.com</pre>

where,

Dynamic agent 3

-gateway local

Indicates that Dynamic Agent 3 can communicate with the dynamic workload broker directly, and a gateway is installed on Dynamic Agent 3 to route communications from dynamic agent workstations that cannot directly communicate with the dynamic workload broker.

-gwid GWADMIN

The gateway ID, GWADMIN, is the name that identifies the gateway on Dynamic Agent 3. Gateways with the same gateway_id can mutually take over in routing communications to the agents connected to them. Specify a different <gateway_id> if the gateways do not communicate with each other.

In addition, configure the two gateways in parallel to take over routing communications from the agents connected to them, should one of the gateways become unavailable. Edit the JobManagerGW.ini file on Dynamic agent 3 and set the JobManagerGWURIs property as follows:

```
JobManagerGWURIs = https://agent3.mycompany.com:31118/ita/JobManagerGW/
JobManagerRESTWeb/JobScheduler/resource,https://agent4.mycompany.com:
31118/ita/JobManagerGW/JobManagerRESTWeb/JobScheduler/resource
```

-tdwbport 31116

The port number of the dynamic workload broker.

-tdwbhostname mainbroker.mycompany.com

The fully qualified host name of the dynamic workload broker.

Dynamic agent 4

-gateway local

Indicates that Dynamic Agent 4 can communicate with the dynamic workload broker directly, and a gateway is installed on Dynamic Agent 4 to route communications from dynamic agent workstations (Dynamic agent 5) that cannot directly communicate with the dynamic workload broker.

-gwid GWADMIN

The gateway ID, GWADMIN, is the name that identifies the gateway site on Dynamic Agent 4. Gateways with the same *<gateway_id>* can mutually take over in routing communications to the agents connected to them. Specify a different *<gateway_id>* if the gateways do not communicate with each other.

In addition, you can configure the two gateways in parallel to take over routing communications from the agents connected to them, should one of the gateways become unavailable. Edit the JobManagerGW.ini file on Dynamic agent 4 and set the JobManagerGWURIs property as follows:

```
JobManagerGWURIs = https://agent3.mycompany.com:31118/ita/JobManagerGW/
JobManagerRESTWeb/JobScheduler/resource,https://agent4.mycompany.com:
31118/ita/JobManagerGW/JobManagerRESTWeb/JobScheduler/resource
```

-tdwbport 31116

The port number of the dynamic workload broker.

-tdwbhostname mainbroker.mycompany.com

The fully qualified host name of the dynamic workload broker.

Dynamic agent 5

-gateway remote

Indicates that Dynamic Agent 5 can connect to the internet through a gateway installed on a different agent, Dynamic Agent 4.

-tdwbport 31118

The port number of the dynamic agent workstation where the gateway resides. In this example, the port number of Dynamic Agent 4 is 31118.

-tdwbhostname agent4.mycompany.com

The fully qualified host name of the dynamic agent workstation where the gateway resides and to which the agent connects.

For information about configuring dynamic agent communications through a gateway, see the *Administration Guide* in the sections Network administration > Network communications.

Configuring a fault-tolerant agent

About this task

After installing a fault-tolerant agent, define the workstation in the database and link the workstation from the master. You can perform this task by using the Dynamic Workload Console or the command line interface. For information, see *User's Guide and Reference*. The following is an example of how to configure a fault-tolerant agent after installation using the command line interface:

1. Log in to the master domain manager as *TWS_user*.
2. Set the environment variables by running `twc_env.sh`.
3. Create the workstation definition in the HCL Workload Automation database. Open a command line window and enter the following commands:

```
composer
new
```

4. Type the workstation definition in the text editor. For example:

```
CPUNAME F235007_00
DESCRIPTION "fault-tolerant agent"
OS UNIX
NODE lab235007
TCPADDR 31111
DOMAIN MASTERDM
FOR MAESTRO
TYPE FTA
AUTOLINK ON
BEHINDFIREWALL OFF
FULLSTATUS OFF
END
```

Run `JnextPlan` with the option **-for 0000** to add the agent workstation definition to the plan and to send the Symphony file to it. For more information about workstation definitions, see the section about workstation definition in *User's Guide and Reference*.



Note: Ensure that the global option carryforward is set to all, otherwise only incomplete job streams are carried forward.

- If you set the autolink parameter to OFF, issue the link command from the master domain manager to link the agent and to download the Symphony file to it:

```
conman "link workstation"
```

- Change the workstation limit to allow jobs to run on the workstation. For example, set the number of jobs to run concurrently on the workstation to 10:

```
conman "limit F235007_00;10"
```

Additionally, the following configuration procedures might be necessary. For information about these procedures, see the relevant sections in *Administration Guide*:

- Customizing and configuring global, local, and user options.
- Customizing and configuring user authentication to allow users authorization on actions and objects, and to configure LDAP.
- Setting connection security to enable SSL for inter-component communications.

Installing additional HCL Workload Automation components

This section describes how to install additional HCL Workload Automation components.

If you need to install more HCL Workload Automation components, for example if you need to add an additional component to an existing installation, you can perform the steps described in the relevant topic:

- [Installing an additional backup domain manager on page 141](#)
- [Installing dynamic domain components on page 144](#)
- [Installing agents on IBM i systems on page 155](#)

Installing an additional backup domain manager

Considerations about installing an additional backup domain manager



You can perform a typical installation, as described in the following scenario, or you can customize the installation parameters, as described in [FAQ - master domain manager and backup master domain manager customizations on page 107](#).

The backup domain manager shares the database with its master domain manager and requires a dedicated Open Liberty, installed on the same workstation as the backup domain manager.

After installing a master domain manager, the administrator runs the **serverinst** command again to install a backup domain manager on a dedicated workstation. The backup domain manager is an agent that can assume the responsibilities of its master domain manager. The **serverinst** command connects to the database you specify, discovers that a master domain manager is already installed, and proceeds to install a backup domain manager.

You might want to install an additional backup domain manager for increased performance and reliability, for example you can move the event processor or the Dynamic Workload Console workload to the backup domain manager.

The HCL Workload Automation administrator needs the following information, which is the same provided when installing the master domain manager, with the exception of the Open Liberty installation directory, which is located on the workstation where you are installing the backup domain manager:

Table 10. Required information

Command parameter	Information type	Provided in...
Database information		
--rdbmstype	database type	Creating and populating the database on page 59
--dbhostname	database hostname	
--dbport	database port	
--dbname	database name	
--dbuser	database user name	
--dbpassword	database password	
HCL Workload Automation information		
--wauser	HCL Workload Automation administrative user name	Creating the HCL Workload Automation administrative user on page 99
--wapassword	HCL Workload Automation administrative user password	
Open Liberty information		
--wlpdir	Open Liberty installation directory	Installing Open Liberty on page 56

Before starting the backup domain manager installation, ensure the following steps have been completed:

1. [Installing Open Liberty on page 56](#) on the workstation where you plan to install the backup domain manager.
2. [Encrypting passwords \(optional\) on page 58](#).
3. [Creating and populating the database on page 59](#) for the master domain manager. The backup domain manager shares the database with the master domain manager.

4. [Creating the HCL Workload Automation administrative user on page 99](#)
5. [Installing the master domain manager and backup master domain manager on page 100](#)
6. On UNIX™ operating systems, ensure that **umask** is set to **022**. To verify that **umask** is set to the correct value, from a command prompt, run the **umask** command. If the value is different from **022**, modify it by running the following command:

```
umask 022
```

You can run the **serverinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. For more information about all **serverinst** parameters and default values, see [Server components installation - serverinst script on page 442](#).

A properties file named `serverinst.properties` is available if you do not want to type parameters in the command line. This is especially useful if you need to specify many parameters or if you want to reuse the file for several installations. The file is located in `image_location/TWS/interp_name`.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

If you need to modify any of the default values, edit the `serverinst.properties` file, but do not modify the `serverinst.template` file located in the same path.

To install the backup domain manager, perform the following steps:

1. Log in to the workstation where you plan to install as root.
2. Browse to the folder where the `serverinst` command is located in `image_location/TWS/interp_name`.
3. . Start the installation specifying a minimum set of parameters. In this case, default values are used for all remaining parameters:

On Windows operating systems

```
cscript serverinst.vbs --acceptlicense yes --rdbmstype <db_type>
--dbhostname <db_hostname> --dbport <db_port> --dbname <db_name>
--dbuser <db_user> --dbpassword <db_password> --wauser <wa_user>
--wapassword <wa_password> --wlpdir <Liberty_installation_dir>
```

On UNIX operating systems

```
./serverinst.sh --acceptlicense yes --rdbmstype <db_type>
--dbhostname <db_hostname> --dbport <db_port> --dbname <db_name>
--dbuser <db_user> --dbpassword <db_password> --wauser <wa_user>
--wapassword <wa_password> --wlpdir <Liberty_installation_dir>
```

4. To verify that the installation completed successfully, browse to the directory where you installed the backup domain manager and type the following commands:

```
. ./twc_env.sh
```

```
optman ls
```

This command lists the HCL Workload Automation configurations settings and confirms that HCL Workload Automation installed correctly.

You have now successfully installed the backup domain manager.

If you want to customize more installation parameters, see [FAQ - master domain manager and backup master domain manager customizations on page 107](#).

Installing dynamic domain components

Procedure to install the dynamic domain manager and backup dynamic domain manager



A dynamic domain manager is the management hub in a domain running both static and dynamic workload. All communications to and from the dynamic agents in the domain are routed through the dynamic domain manager.

The following topics describe the required steps:

1. [Installing Open Liberty on page 144](#)
2. [Encrypting passwords \(optional\) on page 146](#)
3. [Creating and populating the database for the dynamic domain manager on page 148](#)
4. [Creating the HCL Workload Automation administrative user on page 150](#)
5. [Installing the dynamic domain manager and backup dynamic domain manager on page 151](#)

Installing Open Liberty

Open Liberty is required on all workstations where you plan to install the server components and the Dynamic Workload Console.

Before you begin



Ensure that your system meets the operating system and Java requirements. For more information, see Open Liberty detailed system requirements.

About this task

You can quickly install Open Liberty by extracting an archive file on all supported platforms.

If you already have WebSphere Application Server Liberty Base installed, you can use it with HCL Workload Automation, otherwise you can install Open Liberty, as described below.

If you want to move from WebSphere Application Server Liberty Base to Open Liberty, see the topic about moving from WebSphere Application Server Liberty Base to Open Liberty in *Administration Guide*.

Install Open Liberty on all of the following workstations, which comprise a typical installation:

- master domain manager
- backup domain manager
- two Dynamic Workload Console installations on two separate workstations

If you plan to install a dynamic domain manager and its backup, these components require a separate Open Liberty installation.

On UNIX workstations, you can install Open Liberty using a user of your choice. In this case, assign the HCL Workload Automation administrative user read and write access to the Open Liberty installation directory.

To install Open Liberty, perform the following steps:

1. Find out which version of Open Liberty is required, by checking the required version of the Application server in the **Supported Software Report**, available in Product Requirements.
2. Download Open Liberty from [Get started with Open Liberty](#). Download the package named **All GA Features**
3. Perform one of the following actions:
 - a. Extract Open Liberty using the root user:

On Windows operating systems

```
unzip <openliberty_download_dir>\openliberty-<version>.zip
-d <install_dir>
```

On UNIX operating systems

```
unzip <openliberty_download_dir>/openliberty-<version>.zip
-d <install_dir>
```

- b. Run the following command to assign permissions:

```
chmod 755 -R "wlp_directory"
```

OR

Extract Open Liberty using the user who is going to install the product, as follows:

```
su - "wauser"
unzip
```

where:

<openliberty_download_dir>

The directory where you downloaded Open Liberty.

install_dir

The directory where you want to install Open Liberty.



Note: Install the new Open Liberty in the exact location of the previous WebSphere Application Server Liberty Base installation.

4. Ensure the HCL Workload Automation administrative user has the rights to run Open Liberty and full access to the installation directory. If Open Liberty is shared between the master domain manager and the Dynamic Workload Console, ensure also the Dynamic Workload Console user has the same rights.

Results

You have now successfully installed Open Liberty.

What to do next

You can now proceed to [Encrypting passwords \(optional\) on page 146](#).

Encrypting passwords (optional)

How to encrypt passwords required by the installation process

About this task



You can optionally encrypt the passwords that you will use while installing, upgrading, and managing HCL Workload Automation. The secure command uses the AES method and prints the encrypted password to the screen or saves it to a file.



Note: It is important you understand the limits to the protection that this method provides. The custom passphrase you use to encrypt the passwords is stored in clear format in the `passphrase_variables.xml` file, stored in `configureDropin`. To fully understand the implications of this method, it is recommended you read the information provided by Open Liberty at the link [Password encryption limitations](#).

You can perform a typical procedure, which uses a custom passphrase, as described in the following scenario. For more information about all secure arguments and default values, see [Optional password encryption - secure script on page 427](#).

Encrypting the password

1. Browse to the folder where the secure command is located:
 - Before the installation, the command is located in the product image directory, `<image_directory>/TWS/<op_sys>/Tivoli_LWA_<op_sys>/TWS/bin`
 - After the installation, the command is located in `TWA_home/TWS/bin`
2. Depending on your operating system, encrypt the password as follows:

Windows operating systems

```
secure -password password -passphrase passphrase
```

UNIX operating systems

```
./secure -password password -passphrase passphrase
```

z/OS operating systems

```
./secure -password password -passphrase passphrase
```

where

-password

Specifies the password to be encrypted.

-passphrase

Specifies the custom passphrase that is used to generate the key with which the command encrypts the password. If you set this parameter, inform the user who installs HCL Workload Automation that they must define the **SECUREWRAP_PASSPHRASE** environment variable in the same shell from which they run the installation command, and set it to the same value as the **passphrase** parameter. On Windows operating systems, the passphrase must be at least 8 characters long. This argument generates a password which can be reused for all HCL Workload Automation components. This parameter is mutually exclusive with the [-useaeskeystore on page 429](#) parameter, which generates a password which can be decrypted only on the local workstation and not reused for other components.

3. Provide both the encrypted password and custom passphrase to the user in charge of installing HCL Workload Automation. You can use encrypted passwords only in association with the specific passphrase used to encrypt them.

Installing with the encrypted password

The user in charge of installing HCL Workload Automation must set the **SECUREWRAP_PASSPHRASE** environment variable by performing the following steps:

1. Open a brand new shell session.
2. Ensure that no value is set for the **SECUREWRAP_PASSPHRASE** environment variable.
3. Define the **SECUREWRAP_PASSPHRASE** environment variable and set it to the passphrase defined by the user who ran the secure command, as follows:

```
SECUREWRAP_PASSPHRASE=<passphrase>
```

You can use encrypted passwords only in association with the specific passphrase used to encrypt them.

4. In the same shell session, provide the encrypted passwords when running any command that uses a password. An encrypted password looks like the following example:

```
{aes}AFC3jj9cR0YyqR+3CONBzVi8deLb2Bossb9GGroh8UmDPGikIkzXZzid3nzY0IhnSg=
```

What to do next

You can now proceed to [Creating and populating the database for the dynamic domain manager on page 148](#).

Creating and populating the database for the dynamic domain manager

Instructions for creating and populating the HCL Workload Automation database for the dynamic domain manager

About this task



The procedure for creating the database for the dynamic domain manager is identical to that of the master domain manager, with the exception that an additional parameter, `component_type`, must be passed to the script.

For the complete procedure for creating and populating the database, see [Creating and populating the database on page 59](#), then select the procedure related to the database you are using.

To create a DB2 database for the dynamic domain manager submit the following command:

On Windows operating systems

```
cscript configureDb.vbs --componenttype DDM --dbhostname db_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser db_administrator
--dbadminuserpw db_administrator_password
```

On UNIX operating systems

```
./configureDb.sh --componenttype DDM --dbhostname db_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser db_administrator
--dbadminuserpw db_administrator_password
```

To create an Oracle database for the dynamic domain manager submit the following command:

On Windows operating systems

```
cscript configureDb.vbs --componenttype DDM --rdbmstype ORACLE
--dbname service_name --dbuser db_user --dbpassword db_password
--dbhostname db_hostname --dbadminuser db_administrator
--dbadminuserpw db_administrator_password
```

On UNIX operating systems

```
./configureDb.sh --componenttype DDM --rdbmstype ORACLE
--dbname service_name --dbuser db_user --dbpassword db_password
--dbhostname db_hostname --dbadminuser db_administrator
--dbadminuserpw db_administrator_password
```

To create an MSSQL database for the dynamic domain manager submit the following command:

On Windows operating systems

```
cscript configureDb.vbs --componenttype DDM --rdbmstype MSSQL
--dbname db_name --dbhostname db_hostname
--dbadminuser db_administrator
--dbadminuserpw db_administrator_password
```


On UNIX operating systems

```
./configureDb.sh --componenttype DDM --rdbmstype MSSQL
--dbname db_name --dbhostname db_hostname
--dbadminuser db_administrator
--dbadminuserpw db_administrator_password
```

where:

--componenttype

The HCL Workload Automation for which the database is installed. When installing a dynamic domain manager, specify **DDM**.

--dbhostname db_hostname

The host name or IP address of database server.

--dbport db_port

The port of the database server.

--dbname db_name

The name of the HCL Workload Automation database. Note that this name must match the name specified in the `serverinst` command. For more information about the `serverinst` command, see [Server components installation - serverinst script on page 442](#). When creating the database on Oracle, this parameter indicates the service name.

--dbuser db_user

The user that has been granted access to the HCL Workload Automation tables on the database server.

--dbpassword db_password

(Oracle DB only) The password for the user that has been granted access to the HCL Workload Automation tables on the database server. Special characters are not supported.

--dbadminuser db_admin_user

The database administrator user that creates the HCL Workload Automation schema objects on the database server.

--dbadminuserpw db_admin_password

The password of the DB administrator user that creates the HCL Workload Automation schema objects on the database server. Special characters are not supported.

The same criteria apply when creating the database for all supported databases. For more information about creating the database for each supported vendor, see:

- [Creating and populating the database for DB2 for the master domain manager on page 61](#)
- [Creating the database for Oracle and Amazon RDS for Oracle for the master domain manager on page 69](#)
- [Creating the database for MSSQL for the master domain manager on page 74](#)

Results

You have now successfully created and populated the HCL Workload Automation database.

What to do next

You can now proceed to [Creating the HCL Workload Automation administrative user on page 150](#).

Creating the HCL Workload Automation administrative user

Instructions to create the HCL Workload Automation administrative user



HCL Workload Automation administrative user

The HCL Workload Automation administrator creates the administrative user (**wauser**). The administrative user is the user for which the product will be installed in the subsequent steps. This implies that this user has full access to all scheduling objects.

The user name can contain alphanumeric, dash (-), and underscore (_) characters; it cannot contain national characters. The first character of the user name must be a letter.

The following considerations apply:

On Windows operating systems:

- If this user account does not already exist, it is automatically created at installation time.
- If installing on a Windows™ server in a domain, do not define a domain and local ID with the same user name.
- If you specify a domain user, define the name as *domain_name\user_name*.
- If you specify a local user, define the name as *system_name\user_name*. Type and confirm the password.

On UNIX and Linux operating systems:

This user account must be created manually before running the installation and must be enabled to login to the machine where the master domain manager is going to be installed. Create a user with a home directory and group. Use the appropriate UNIX and Linux operating system commands to create the user.



Important: Group names that contain a "/" (forward slash) character can cause permissions to not be set correctly. When HCL Workload Automation retrieves credentials from WebSphere Application Server Liberty, it parses the returned list of groups names assuming they are saved in the format `<realm_name>/<group_name>`. If the group name, the realm name, or both contain a "/" character, the parsing fails.

You can also install HCL Workload Automation using a user different from the root user. This installation method is known as **no-root installation** and applies to all HCL Workload Automation components. Note that

if you choose this installation method, only the user who performs the installation can use HCL Workload Automation. For this reason, the typical installation scenario described in this section uses the root user.

For more information, see [HCL Workload Automation user management on page 49](#).

Results

You have now successfully created the HCL Workload Automation administrative user.

What to do next

You can now proceed to [Installing the dynamic domain manager and backup dynamic domain manager on page 151](#).

Installing the dynamic domain manager and backup dynamic domain manager

Considerations about installing the dynamic domain manager and backup dynamic domain manager



A dynamic domain manager is the management hub in a domain running both static and dynamic workload. All communications to and from the dynamic agents in the domain are routed through the dynamic domain manager.

You can perform a typical installation, as described in the following scenario, or you can customize the installation parameters, as described in [FAQ - dynamic domain manager customizations on page 154](#). For example, you can install the dynamic domain manager and backup dynamic domain manager using custom certificates.

The dynamic domain manager and backup dynamic domain manager require a dedicated database and a dedicated Open Liberty.

The procedure to install the dynamic domain manager and backup dynamic domain manager is exactly the same, with the difference that it is performed on two different workstations and that each installation points to its local Open Liberty installation. HCL Workload Automation determines whether or not a dynamic domain manager is already present in the environment and proceeds to install a dynamic domain manager or backup dynamic domain manager accordingly.

The HCL Workload Automation administrator installs the dynamic domain manager and backup dynamic domain manager. He needs the following information:

Table 11. Required information

Command parameter	Information type	Provided in...
Database information		

Table 11. Required information

(continued)

--rdbmstype	database type	Creating and populating the database for the dynamic domain manager on page 148
--dbhostname	database hostname	
--dbport	database port	
--dbname	database name	
--dbuser	database user name	
--dbpassword	database password	
HCL Workload Automation information		
--wauser	HCL Workload Automation administrative user name	Creating the HCL Workload Automation administrative user on page 150 and Installing the master domain manager and backup master domain manager on page 100
--wapassword	HCL Workload Automation administrative user password	
--master	The master domain manager name	
--mdmbrokerhostname	The fully qualified host name or IP address of the master domain manager contacted by the dynamic domain manager.	
--mdmhttpsport	The port of the master domain manager host used by the broker to contact master domain manager.	
Open Liberty information		
--wlpdir	Open Liberty installation directory	Installing Open Liberty on page 144

Before starting the installation, ensure the following steps have been completed:

1. [Installing Open Liberty on page 144](#) on the workstation where you plan to install the dynamic domain manager and on the workstation where you plan to install the backup dynamic domain manager.
2. [Encrypting passwords \(optional\) on page 146](#).
3. [Creating and populating the database for the dynamic domain manager on page 148](#)
4. [Creating the HCL Workload Automation administrative user on page 150](#)

- On UNIX™ operating systems, ensure that **umask** is set to **022**. To verify that **umask** is set to the correct value, from a command prompt, run the **umask** command. If the value is different from **022**, modify it by running the following command:

```
umask 022
```

You can run the **serverinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. For more information about all **serverinst** parameters and default values, see [Server components installation - serverinst script on page 442](#).

A properties file named `serverinst.properties` is available if you do not want to type parameters in the command line. This is especially useful if you need to specify many parameters or if you want to reuse the file for several installations. The file is located in `image_location/TWS/interp_name`.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

If you need to modify any of the default values, edit the `serverinst.properties` file, but do not modify the `serverinst.template` file located in the same path.

To install the dynamic domain manager, perform the following steps:

- Log in to the workstation where you plan to install.
- Browse to the folder where the `serverinst` command is located:

On Windows operating systems

```
image_location\TWS\interp_name
```

On UNIX operating systems

```
image_location/TWS/interp_name
```

- Start the installation specifying a typical set of parameters. In this case, default values are used for all remaining parameters:

On Windows operating systems

```
cscript serverinst.vbs --acceptlicense yes --rdbmstype db_type
--dbhostname db_hostname --dbport db_port --dbname db_name
--dbuser db_user --dbpassword db_password --wauser wa_user
--wapassword wa_password --componenttype DDM --domain domain_name
--master mdm_name --mdmbrokerhostname mdm_broker_host_name
--mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir\wlp
```

On UNIX operating systems

```
./serverinst.sh --acceptlicense yes --rdbmstype db_type
--dbhostname db_hostname --dbport db_port --dbname db_name
--dbuser db_user --dbpassword db_password --wauser wa_user
--wapassword wa_password --componenttype DDM --domain domain_name
--master mdm_name --mdmbrokerhostname mdm_broker_host_name
--mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir/wlp
```

Repeat the same procedure on the workstation where you plan to install the backup dynamic domain manager

Result

You have now successfully installed the dynamic domain manager and backup dynamic domain manager.

For more information about all **serverinst** parameters and default values, see [Server components installation - serverinst script on page 442](#).

FAQ - dynamic domain manager customizations

A list of questions and answers related to the customization of the dynamic domain manager installation

When installing the dynamic domain manager, you can perform a typical installation, as described in [Installing the dynamic domain manager and backup dynamic domain manager on page 151](#) or you can customize a number of parameters, as described in the following scenario:

- [How do I install the dynamic domain manager using custom certificates? on page 154](#)

How do I install the dynamic domain manager using custom certificates?

Installing the dynamic domain manager and its backup using custom certificates

About this task

You can install the dynamic domain manager and its backup using default certificates, as described in [Installing the dynamic domain manager and backup dynamic domain manager on page 151](#), or you can optionally use custom certificates.

To install dynamic domain manager and backup dynamic domain manager using custom certificates, perform the following steps:

1. Generate the custom certificates required for installing the dynamic domain manager and backup dynamic domain manager, as follows:

```
openssl genrsa -des3 -out tls.key 2048
```

Result

The following files are created:

- ca.crt
 - tls.key
 - tls.crt
2. Copy the files to a path of your choice on the workstation where you plan to install the dynamic domain manager or backup dynamic domain manager. When performing the installation, you provide this path using the `sslkeysfolder` parameter.
 3. Copy the `tls.crt` file from the master domain manager to the workstation where you plan to install the dynamic domain manager or backup dynamic domain manager. Specify a different path from the path of the above certificates to avoid overwriting the existing `tls.crt`.
 4. Rename the `tls.crt` file from the master domain manager to `jwt.crt`.

5. Copy the `jwt.crt` file to the same path as the certificates generated in step 1.

Result

You now have on the workstation where you plan to install the dynamic domain manager or backup dynamic domain manager four certificate files:

- a. `ca.crt`
- b. `tls.key`
- c. `tls.crt`
- d. `jwt.crt`

6. Browse to the folder where the `serverinst` command is located:

On Windows operating systems

```
image_location\TWS\interp_name
```

On UNIX operating systems

```
image_location/TWS/interp_name
```

7. Start the installation specifying the path to the dynamic domain manager certificates using the `sslkeysfolder` parameter:

On Windows operating systems

```
cscript serverinst.vbs --acceptlicense yes --rdbmstype db_type
--dbhostname db_hostname --dbport db_port --dbname db_name
--dbuser db_user --dbpassword db_password --wauser wa_user
--wapassword wa_password --componenttype DDM --domain domain_name
--master mdm_name --mdmbrokerhostname mdm_broker_host_name
--mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir\wlp
--sslkeysfolder path_to_certificates --sslpassword certificate_password
```

On UNIX operating systems

```
./serverinst.sh --acceptlicense yes --rdbmstype db_type
--dbhostname db_hostname --dbport db_port --dbname db_name
--dbuser db_user --dbpassword db_password --wauser wa_user
--wapassword wa_password --componenttype DDM --domain domain_name
--master mdm_name --mdmbrokerhostname mdm_broker_host_name
--mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir/wlp
--sslkeysfolder path_to_certificates --sslpassword certificate_password
```

8. Repeat the same procedure for the backup dynamic domain manager.

Results

You have now successfully installed the dynamic domain manager and backup dynamic domain manager

Installing agents on IBM i systems

Learn how to install agents on IBM i systems.

About this task

To install dynamic agents and z-centric agents on IBM i systems, use the `twsinst` installation script.

You can either use the default **QSECOFR** user or create a new user with **ALLOBJ** authority. If you plan to use a user different from **QSECOFR**, create the user before the installation and assign it the ALLOBJ authority.

For dynamic agents, you can also use a user **different from QSECOFR** with no specific authority. Note that only the user who performs the installation can use the agent.

Verify that the user profile used as **TWSUser** is not a member of a group profile. Set the group profile associated with the **TWSUser** to ***NONE**. If the **TWSUser** is a member of a group, the installation might fail.

If you plan to enable FIPS, ensure your certificates meet FIPS standards before getting started.

To ensure FIPS compliance, all HCL Workload Automation components must be at version 10.2.5 or later, certificates must employ at least a robust 2K RSA key and use encryption algorithms different from **MD5-RSA** and **SHA1-RSA**.

To install the agents, perform the following steps:

1. Sign on as the user of your choice, either **QSECOFR** or an **existing user with ALLOBJ authority**. If you use a user different from **QSECOFR**, specify the **allobjAuth** parameter to indicate that the specified user has the ALLOBJ authority. Ensure the user is existing and has ALLOBJ authority because the product does not verify that the correct authority is assigned. For more information about the **allobjAuth** parameter, see [Agent installation parameters on IBM i systems on page 159](#). Only for dynamic agents, you can also use a user different from **QSECOFR** with no specific authority.
2. Create an IBM i user profile for which the HCL Workload Automation agent is installed.



Note: The user profile is not the same as for the user that is performing the installation, unless you use a user different from **QSECOFR** with no specific authority (dynamic agents only). Instead the user profile is for the user that you specify in the **-uname username** parameter when running the twsinst script. For descriptions of the syntax parameters, see [Agent installation parameters on IBM i systems on page 159](#). You cannot use an existing IBM i system user profile, an application supplied user profile, or any of the following reserved IBM i user profiles:

- QDBSHR
- QDFTOWN
- QDOC
- QLPAUTO
- QLPINSTALL
- QRJE
- QSECOFR
- QSPL
- QSYS
- QTSTRQS



Attention: Consider that:



- If the user profile is a member of a group, the installation fails. Set the group profile that is associated with the user profile to **NONE*.
- If the *username* is longer than 8 characters, after the installation the agent (and the JobManager component) runs under the **QSECOFR** user instead of under the authority of the installation user. To prevent this problem, set the `PASE_USRGRP_LIMITED` environment variable to N.

3. On the IBM i system, verify that no library exists with the same name as the user profile supplied for the agent user.
4. Download the installation images from [HCL Software](#).
5. To untar or unzip the agent image, you can use the *PASE* shell or the *AIXterm* command.

Using *PASE* shell:

- a. Open the *PASE* shell.
- b. Run the command "CALL QP2TERM".
- c. Locate the folder where you downloaded the agent image and run the command:

HCL Workload Automation Agent

```
"tar xvf TWSversion_number>_IBM_I.tar"
```

Dynamic Agent

```
"unzip TWSversion_number>_IBM_I.zip"
```

- d. Exit from the *PASE* shell.

Using *AIXterm* command:

- a. Start the *Xserver* on your desktop.
- b. On the iSeries machine, open a *QSH shell* and export the display.
- c. In *QSH shell*, go to the directory */QopenSys* and run the command "aixterm -sb".
- d. A pop-up window is displayed on your desktop. By Using this pop-up window, unzip the *TWSversion_number>_IBM_I.zip* file, or untar the *TWSversion_number>_IBM_I.tar* file.

6. If your machine's primary language is other than English, carry out these steps:
 - a. Add English as secondary language.
 - b. Ensure that when connecting to the environment the Host Code-Page is set to 037
 - c. Before starting the installation, verify that the *Qshell session* is configured correctly and type the following command in the <yourfilename> :

```
echo " key key2 " | sed 's/ *$//g' | sed 's/^ */g'
```

- d. Run the <yourfilename>
 - e. The environment is configured in the correct way if the output is: "key key2".
7. Open a *QSH shell* and run the *twinst* script. During the installation process, the product creates an IBM i library and a job description with the same name as the user profile created in [Step 2 on page 156](#).

The installation procedure adds this library to the user profile library list of the dynamic agent user profile and sets this job description as the job description of the dynamic agent user profile. By default, the software is installed in the user's home directory.

If the installation fails to understand the cause of the error, see [Analyzing return codes for agent installation, upgrade, restore, and uninstallation on page 400](#).

After a successful installation, perform the following configuration task:

- [Configuring a dynamic agent on page 216](#), as described in *HCL Workload Automation: Planning and Installation*.

Command usage and version

Show command usage and version

```
twswinst -u | -v
```

Install a new instance

```
twswinst -new -uname username
      -acceptlicense yes|no
      [-addjruntime true|false]
      [-agent dynamic]
      [-allObjAuth]
      [-company company_name]
      [-displayname agentname]
      [-gateway local|remote|none]
      [-gweifport gateway_eif_port]
      [-gwid gateway_id]
      [-hostname hostname]
      [-inst_dir install_dir]
      [-jimport port_number]
      [-jimportssl true|false]
      [-lang lang_id]
      [-tdwbport tdwbport_number]
      [-tdwbhostname host_name]
      [-work_dir working_dir]
```

For a description of the installation parameters and options that are related to agent on this operating system, see [Agent installation parameters on IBM i systems on page 159](#) in *HCL Workload Automation: Planning and Installation*.

Prerequisites

About this task

To install and use the IBM i agent you must have a supported version of the IBM i operating system. For a detailed list of supported operating systems, see the Detailed System Requirements document at [HCL Workload Automation Detailed System Requirements](#).

Scanning system prerequisites on IBM i systems

Scanning system prerequisites on IBM i systems

About this task

Before you install or upgrade the agent, HCL Workload Automation automatically runs a scan on your system. Having an environment that meets the product system requirements ensures that the installation or upgrade succeeds without any delays or complications.

The scan verifies that:

- The operating system is supported for the product.
- There is enough permanent and temporary disk space to install both the product and its prerequisites.
- There is enough memory and virtual memory swap space.



Note: The scan verifies only that the environment meets the requirements of HCL Workload Automation.

If any of these checks fails, HCL Workload Automation performs the following action:

- An error message is returned. Analyze the log file, solve the error, and rerun the installation or upgrade. The log file is in `%TEMP%\TWA\tws1025\result.txt`
- You can decide to rerun the installation or upgrade without executing the prerequisite scan. If you specify the **-skipcheckprereq** parameter, the `twinst` installation script does not execute the prerequisite scan. For more information about the `-skipcheckprereq` option, see [Agent installation parameters - twinst script on page 119](#).

For a detailed list of supported operating systems and product prerequisites, see [HCL Workload Automation Detailed System Requirements](#).

Agent installation parameters on IBM i systems

About this task

The parameters set when using the **twinst** script to install dynamic and z-centric agents on IBM i systems.

-acceptlicense yes/no

Specifies whether to accept the License Agreement.

-addjruntime true/false

Adds the Java™ run time to run job types with advanced options, both those types that are supplied with the product and the additional types that are implemented through the custom plug-ins. Valid values are **true** and **false**. The default for a fresh installation is **true**. Set this parameter to `true` if you use the **sslkeysfolder** and **sslpassword** parameters to define custom certificates in PEM format.

If you decided not to install Java™ run time at installation time, you can still add this feature later as it is described in [Adding a feature on page 225](#).

-allObjAuth

If you are installing, upgrading, or uninstalling with a user different from the default **QSECOFR** user, this parameter specifies that the user has the required ALLOBJ authority. Ensure the user is existing and has

ALLOBJ authority because the product does not verify that the correct authority is assigned. The same user must be specified when installing, upgrading or uninstalling the agent. If you are using the **QSECOFR** user, this parameter does not apply.

-apikey

Specifies the API key for authentication with the master domain manager. This key enables downloading certificates or JWT for communication between dynamic agent and dynamic domain manager. A random password in base64 encoding is automatically created for generating stash files. The password stored in the `tls.sth` file. If needed, you can decrypt this password using any base64 decoder.

Obtain the string to be provided with this parameter from the Dynamic Workload Console before running the command. For more information, see the section about authenticating the command line client using API Keys in *Dynamic Workload Console User's Guide*.

This parameter is **mutually exclusive** with:

- [-wauser wauser_name on page 129](#)
- [-wapassword wauser_password on page 129](#)
- [-sslkeyfolder path on page 126](#)
- [-sslpassword password on page 127](#)

and it is **required** with:

- [-tdwbhostname host_name on page 127](#)
- [-tdwbport tdwbport_number on page 127](#)

For a comprehensive list of supported combinations for this parameter and related ones, see [Table 7: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 130](#).

-company company_name

The name of the company. The company name cannot contain blank characters. The name is shown in program headers and reports. If not specified, the default name is COMPANY.

-displayname display_name

The name to assign to the agent. The name cannot start with a number. The default is based on the host name of this computer.

If the host name starts with a number, the **-displayname** parameter must be specified.

-enablefips true/false

Specify whether you want to enable FIPS. The default value is `false`. This parameter is optional.

-gateway *local/remote/none*

Specifies whether to configure a gateway to communicate with the dynamic workload broker or not, and how it is configured. Specify *local* if the gateway is local to the dynamic agent workstation. Specify *remote* if the dynamic agent communicates through a gateway that is installed on a different dynamic agent workstation from the dynamic agent being installed. The default value is *none*, which means no gateway is configured. For information about installing with a local and remote gateway, see [Example installation commands on page 132](#).

-gweifport *gateway_elf_port*

Specifies the Job Manager Event Integration Facility (EIF) port number. The default value is **31132**. The valid range is 1 to 65535.

-gwid *gateway_id*

The unique identifier for the gateway. This parameter is required when you specify **-gateway *local*** and must be unique across all agents. The default gateway identifier that is assigned is **GW1**. The gateway identifier must start with either an alphabetic character or an underscore character (_), and it can contain only the following types of characters: alphabetic, numeric, underscores (), hyphens (-), and periods (.).

Gateways can also work in parallel to mutually take over in routing communications to the agents connected to them. To enable gateways to work in parallel, all gateways must have the same *gateway_id* assigned. This information is stored in the `JobManagerGW.ini` file, by setting the **JobManagerGWURLs** property.

-hostname *host_name*

The fully qualified hostname or IP address on which the agent is contacted by the dynamic workload broker. The default is the hostname of this computer. If the hostname is a localhost, the hostname parameter must be specified.

-inst_dir *installation_dir*

The directory of the HCL Workload Automation installation. Specify an absolute path. The path cannot contain blanks. If you do not manually specify a path, the path is set to the default home directory, that is, the *home/username* directory, where *username* is the value specified in the `-uname` option.

-jimport *port_number*

The JobManager port number used by the dynamic workload broker to connect to the dynamic agent. The default value is **31114**. The valid range is from 1 to 65535.

-jimportssl *true/false*

The JobManager port used by the dynamic workload broker to connect to the HCL Workload Automation dynamic agent. The port value is the value of the `ssl_port` parameter in the `ita.ini` file if **-jimportssl** is set to *true*. If set to *false*, it corresponds to the value of the **tcp_port** parameter in the `ita.ini` file. The `ita.ini` file is located in `ITA\cpa\ita` on Windows™ systems and `ITA/cpa/ita` on UNIX™, Linux™, and IBM i systems.

Set the value to "true" if **- gateway** is set to *local*.

For communication using SSL or HTTPS

Set **jimportssl = true**. To communicate with the dynamic workload broker, it is recommended that you set the value to *true*. In this case, the port specified in **jimport** communicates in HTTPS.

For communication without using SSL or through HTTP

Set **jimportssl = false**. In this case the port specified in **jimport** communicates in HTTP.

-lang lang_id

The language in which the twsinst messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used. If neither **-lang** nor LANG are used, the default codepage is set to SBCS. For a list of valid values for these variables, see the following table:

Table 12. Valid values for -lang and LANG

parameter

Language	Value
Brazilian portuguese	pt_BR
Chinese (traditional and simplified)	zh_CN, zh_TW
English	en
French	fr
German	de
Italian	it
Japanese	ja
Korean	ko
Russian	ru
Spanish	es



Note: This is the language in which the installation log is recorded and not the language of the installed engine instance. twsinst installs all languages as default.

-new

A fresh installation of the agent. Installs an agent and all supported language packs.

-skip_usercheck

Enable this option if the authentication process within your organization is not standard, thereby disabling the default authentication option. If you specify this parameter, you must create the user manually before running the script.

-skipcheckprereq

If you specify this parameter, HCL Workload Automation does not scan system prerequisites before installing the agent.

For a detailed list of supported operating systems and product prerequisites, see [HCL Workload Automation Detailed System Requirements](#).

-sslkeysfolder

The name and path of the folder containing the certificates in PEM format. The installation program generates the keystore and truststore files using the password you specify with the **--sslpassword** parameter. If you use this parameter, ensure that the **addjruntime** parameter is set to true, because Java™ run time is required for defining custom certificates. This parameter is not supported on HCL Workload Automation Agent (also known as the agent with z-centric capabilities).

-sslpassword

Specify the password for the certificates automatically generated by the installation program. If you use this parameter, ensure that the **addjruntime** parameter is set to true, because Java™ run time is required for defining custom certificates. This parameter is not supported on HCL Workload Automation Agent (also known as the agent with z-centric capabilities).

-tdwbhostname *host_name*

The fully qualified host name of the dynamic workload broker. It is used together with the **-agent *dynamic*** and the **-tdwbport *tdwbport_number*** parameters. This value is registered in the **ResourceAdvisorUrl** property in the `JobManager.ini` file. This parameter is required if you use the **wauser** and **wapassword** or the **apikey** parameters.

If you set the **-gateway** parameter to `remote`, this is the host name of the dynamic agent hosting the gateway and to which the agent you are installing will connect. This information is stored in the `JobManager.ini` file. For information about installing with a local and remote gateway, see [Example installation commands on page 132](#).

-tdwbport *tdwbport_number*

The dynamic workload broker HTTP or HTTPS transport port number. It is used together with the **-agent *dynamic*** and the **-tdwbhostname *host_name*** parameters. The valid range is from 0 to 65535. If you specify **0**, you cannot run workload dynamically. Do not specify **0** if the **-agent** value is **dynamic**. This number is registered in the **ResourceAdvisorUrl** property in the `JobManager.ini` file. This parameter is required if you use the **wauser** and **wapassword** or the **apikey** parameters.

If you set the **-gateway** parameter to `remote`, this is the HTTP or HTTPS port number of the dynamic agent hosting the gateway and to which the agent you are installing will connect. You have specified this port with the **jimport** parameter when installing the agent hosting the gateway. For information about installing with a local and remote gateway, see [Example installation commands on page 132](#).

-thiscpu workstation

The name of the HCL Workload Automation workstation of this installation. The name cannot exceed 16 characters, cannot start with a number, cannot contain spaces, and cannot be the same as the workstation name of the master domain manager. This name is registered in the `localopts` file. If not specified, the default value is the host name of the workstation.

If the host name starts with a number, **-thiscpu** parameter must be specified.

-u

Displays command usage information and exits.

-uname username

The name of the user for which HCL Workload Automation is installed.

If you are using the **QSECOFR** user or a user with **ALLOBJ authority**, this user name is not the same as the user performing the installation. If you are using a user **different from QSECOFR**, the user performing the installation and the user for which the agent is installed are the same.

If *username* is longer than 8 characters, after installation the agent (and the JobManager component) erroneously run under the **QSECOFR** user, instead of under the authority of the installation user. To prevent this, set the `PASE_USRGRP_LIMITED` environment variable to N.

-v

Displays the command version and exits.

-wapassword wouser_password

One of the following passwords, defined on the master domain manager:

- The password of the user for which you have installed the master domain manager the agent is connecting to.
- The password of the user with the DISPLAY permission on the FILE named AGENT_CERTIFICATE. This permission allows the user to download certificates or JWT. For more information about this scenario, see [Downloading certificates or JWT using a different user on page 487](#).

Always specify the user defined on the master domain manager, also if you are installing a dynamic agent and want it to register to a dynamic domain manager. This is because the dynamic domain manager simply forwards data to and from the master domain manager.

By providing the **wouser** and **wapassword** parameters or the **apikey** parameter, you enable HCL Workload Automation to download and install either the certificates or the JWT already available on the master domain manager:

See also [-jwt true | false on page 124](#).

Key details about this parameter:

- It is **mutually exclusive** with the [-apikey on page 121](#) parameter, which provides authentication using an API Key and the [-sslkeysfolder path on page 126](#) and [-sslpassword password on page 127](#) parameters.
- It **always requires** the [tdwbport on page 127](#) and [-tdwbhostname host_name on page 127](#) parameters.
- It is **not supported** on the HCL Workload Automation Agent (also known as the agent with z-centric capabilities). To generate certificates for the HCL Workload Automation Agent, use the **sslkeysfolder** and **sslpassword** parameters.

You can optionally encrypt the password using the secure script. For more information, see [Optional password encryption - secure script on page 427](#).

For further information about how to automatically download and deploy certificates in PEM format from the master domain manager to dynamic agents and fault-tolerant agents, see [Certificates download to dynamic agents and fault-tolerant agents - AgentCertificateDownloader script on page 482](#).

This parameter always requires the [tdwbport on page 127](#) and [-tdwbhostname host_name on page 127](#) parameters.

For a comprehensive list of supported combinations for this parameter and related ones, see [Table 7: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 130](#).

-wauser wauser_name

One of the following users, defined on the master domain manager:

- The user for which you have installed the master domain manager the agent is connecting to.
- The user with the DISPLAY permission on the FILE named AGENT_CERTIFICATE. This permission allows the user to download certificates or JWT. For more information about this scenario, see [Downloading certificates or JWT using a different user on page 487](#).

Always specify the user defined on the master domain manager, also if you are installing a dynamic agent and want it to register to a dynamic domain manager. This is because the dynamic domain manager simply forwards data to and from the master domain manager.

By providing the **wauser** and **wapassword** parameters or the **apikey** parameter, you enable HCL Workload Automation to download and install either the certificates or the JWT already available on the master domain manager:

- To download certificates, set the **jwt** parameter to `false`
- To download JWT, set the **jwt** parameter to `true`. For more information, see [-jwt true | false on page 124](#).

Key details about this parameter:

- It is **mutually exclusive** with the [-apikey on page 121](#) parameter, which provides authentication using an API Key and the [-sslkeysfolder path on page 126](#) and [-sslpassword password on page 127](#) parameters.
- It **always requires** the [tdwbport on page 127](#) and [-tdwbhostname host_name on page 127](#) parameters.
- It is **not supported** on the HCL Workload Automation Agent (also known as the agent with z-centric capabilities). To generate certificates for the HCL Workload Automation Agent, use the **sslkeysfolder** and **sslpassword** parameters.

For further information about how to automatically download and deploy certificates in PEM format from the master domain manager to dynamic agents and fault-tolerant agents, see [Certificates download to dynamic agents and fault-tolerant agents - AgentCertificateDownloader script on page 482](#).

For a comprehensive list of supported combinations for this parameter and related ones, see [Table 7: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 130](#).

-work_dir working_dir

The temporary directory used for the HCL Workload Automation installation process files deployment.

The path cannot contain blanks. If you do not manually specify a path, the path is set to `/tmp/TWA/twsversion_number>`.

Example installation of an agent on IBM i systems

About this task

The following example shows the syntax used when using the **twsinst** script to install a new instance of the agent on an IBM i system.

```
./twsinst -new
-uname TWS_user
-acceptlicense yes
-hostname thishostname.mycompany.com
-jmport 31114
-tdwbport 41114
-tdwbhostname mainbroker.mycompany.com
-work_dir "/tmp/TWA/tws93"
```

The twsinst script log files on IBM i systems

About this task

The twsinst log file name is:

Where: `<TWS_INST_DIR>/twsinst_IBM_i_TWS_user^product_version.log`

TWS_INST_DIR

The HCL Workload Automation installation directory. The default installation directory is `/home/TWS_user`.

TWS_user

The name of the user for which HCL Workload Automation was installed, that you supplied during the installation process.

product_version

Represents the product version. For example, for version 10.2.5 of the product, the value is 10.2.5.00

Chapter 6. Deploying with containers

Deploy HCL Workload Automation quickly and easily with containers.

Following you can find more details about the HCL Workload Automation deployment with containers based on your environment.

Docker containers

An easy and fast deployment method of HCL Workload Automation. Docker compose is a method to instantly download the product image, create a container, and start up the product.

Docker is a state-of-the-art technology which creates, deploys, and runs applications by using containers. Packages are provided containing an application with all of the components it requires, such as libraries, specific configurations, and other dependencies, and deploy it in no time on any other Linux or Windows workstation, regardless of any different settings between the source and the target workstation.

Docker adoption ensures standardization of your workload scheduling environment and provides an easy method to replicate environments quickly in development, build, test, and production environments, speeding up the time it takes to get from build to production significantly. Install your environment using Docker to improve scalability, portability, and efficiency.

Docker containers are available for UNIX, Windows and Linux on Z operating systems.

For more information, see the introductory readme file for all components available at [HCL Workload Automation](#). You can also find detailed information for each component in the related readme file, as follows:

- [HCL Workload Automation Server](#)
- [HCL Workload Automation Console](#)
- [HCL Workload Automation dynamic agent](#)
- [HCL Workload Automation z-centric agent](#)
- [Workload Automation FileProxy](#)

You can also use docker containers to store all the latest integrations available on Automation Hub. For further information see: [Container plug-in](#).

Amazon Web Services (AWS) Marketplace

You can use Amazon Web Services (AWS) Marketplace to subscribe to HCL Workload Automation and deploy your environment on the AWS secure cloud platform.

For more information see [Deploying from Amazon Web Services \(AWS\) Marketplace on page 176](#).

Amazon Web Services (AWS) Elastic Kubernetes Service (EKS) (Amazon EKS)

You can use Amazon EKS to run HCL Workload Automation containerized product components on the Amazon Web Services secure cloud platform.

For more information, see [Deploying on Amazon EKS on page 184](#)

Azure Kubernetes Service (AKS)

Deploy and manage HCL Workload Automation containerized product components on the Azure AKS, a container orchestration service available on the Microsoft Azure public cloud. You can use Azure AKS to deploy, scale up, scale down and manage containers in the cluster environment. You can also deploy and run an Azure SQL database.

For more information, see [Deploying on Azure AKS on page 185](#).

Google GKE

Google Kubernetes Engine (GKE) provides a managed environment for deploying, managing, and scaling your containerized applications using Google infrastructure. The Google GKE environment consists of multiple machines grouped together to form a cluster. You can also deploy and run Google Cloud SQL for SQL server.

Google GKE supports session affinity in a load balancing cluster, a feature which maintains each user session always active on the same pod. This ensures that the Dynamic Workload Console always connects to the same server during a session and that the user can perform any number of operations smoothly and seamlessly.

For more information, see [Deploying on Google GKE on page 185](#).

Red Hat OpenShift

You can deploy the HCL Workload Automation components using IBM® certified containers. For further information, see [Deploying HCL Workload Automation components on Red Hat OpenShift using helm charts on page 176](#).

All HCL Workload Automation product components can be deployed on Red Hat OpenShift, V4.x. Red Hat OpenShift is a container application platform based on Kubernetes to orchestrate containers.

Considerations about deploying with containers

Some considerations about your HCL Workload Automation environments when the product components are deployed using containers.

An environment deployed with containers has some characteristics that differ from an environment installed using the classic installation method. Following a list of its characteristics:

- Container deployment is supported only for dynamic agents and not for fault-tolerant agents, and external fault-tolerant agents are not supported on Kubernetes.
- All dynamic agents must obligatorily be configured to use a dynamic agent gateway.
- Each time a `switcheventprocessor` command is issued, a `switchmgr` command must also be issued on the same node.
- An on-premises fault-tolerant agent cannot connect to a master domain manager for on-cloud solutions supported by HCL Workload Automation (only dynamic agents are supported).
- The HCL Workload Automation event processor service must run on the same machine where the current master is running because the host and port are re-mapped by dynamic agents to use the master server host and port. Thus, performing a switch from master to backup master, you must also switch the event processor on the new master.

- The console, server and agent components are installed with non-root user (`wauser`) that does not include sudoers privileges. This implies that jobs that run on agents on containers can run only with `wauser` user and cannot impersonate other users.
- An extended agent component, (`RELEASE_NAME-waserver-0_XA`), is automatically created on the server. It starts the scheduling process by running the FINAL job stream that generates the daily production plan.
- An FTA container is not provided (only dynamic agents are supported in containers).
- By default, the FINAL job stream has a start time of 07:00 and invokes MAKEPLAN at 07:00. The Start of Day is 00:00. MAKEPLAN extends the plan until 09:00 the following day. If you modify the scheduling time of the FINAL job stream to a time different from the default, then evaluate whether you should also manually modify the plan extension time defined in the MAKEPLAN job accordingly.

The following is an example of the default output when you run `planman showinfo`:

```
Locale LANG set to the following: "en"
Plan creation start time: 06/23/2020 00:00 TZ UTC
Production plan start time of last extension: 06/24/2020 09:00 TZ UTC
Production plan end time: 06/25/2020 08:59 TZ UTC
Production plan time extension: 024:00
Plan last update: 06/24/2020 07:00 TZ UTC
Preproduction plan end time: 07/08/2020 00:00 TZ UTC
Start time of first not complete preproduction plan job stream instance: 06/23/2020 00:00 TZ UTC
```

Customizing container parameters

The document describes how to avoid the overwriting of the customized parameters added in the container configuration files, such as the `datasource.xml` file.

It is possible to customize the container configuration by adding parameters, for example, in the `datasource.xml` file that is located in the following path:

```
/opt/wautils/dropins
```

Restarting a container, the `datasource.xml` file is overwritten and the customized parameters inside it are lost; to avoid that, proceed as follows:

- Create another `.xml` file with a name that is listed in a higher alphabetical order than `datasource.xml`.
- In the new `.xml` file, add the parameters to be customized together with the corresponding section.

In this way - at the restart of the container - the customized parameters are not overwritten.

Deploying with Docker compose

Getting started with Docker compose

This topic gives you an overview of the high-level procedure to deploy HCL Workload Automation components using Docker.

To deploy HCL Workload Automation using a Docker container, proceed as follows:

1. Ensure that all of the prerequisites are met as documented in [Prerequisites on page 171](#). If you are deploying on Linux on Z, ensure you perform the preparatory steps documented in [Deploying Docker compose on Linux on Z on page 171](#).
2. Access and then download the Docker image from the entitled registry. For further information, see the complete procedure in [Deploying containers with Docker on page 173](#).
3. You can choose to deploy all product containers with a single command, or you can deploy each product component container individually.

For more information, see the introductory readme file for all components available at [HCL Workload Automation](#).

You can also find detailed information for each component in the related readme file, as follows:

- [HCL Workload Automation Server](#)
- [HCL Workload Automation Console](#)
- [HCL Workload Automation dynamic agent](#)
- [HCL Workload Automation z-centric agent](#)
- [Workload Automation FileProxy](#)

4. Access the container to verify the status and run HCL Workload Automation commands. For further details see [Accessing the Docker containers on page 175](#).

Prerequisites

Prerequisite information when deploying with containers.

When deploying the product using containers, ensure you have fulfilled the following prerequisites:

Check the prerequisites of the command-line installation method in [Prerequisites on page 45](#).

If you want to calculate the necessary resources that the agent container needs to run, use the following formula:

Evaluate the volume_size variable:

```
Volume size(MB)=
    120 + [ 30 x jobs_per_day x (average_joblog_size_MB / 3 + 0.008) ]
```

For example, considering "average_joblog_size_MB = 0.001 MB (1KB)", you obtain:

```
1.000
    jobs_per_day: 370 MB --> volume_size = 370Mi
```

```
10.000
    jobs_per_day: 2.6 GB --> volume_size = 2600Mi
```

```
100.000
    jobs_per_day: 25 GB --> volume_size = 25Gi
```

Deploying Docker compose on Linux on Z

Before you deploy HCL Workload Automation components on Linux on Z, ensure that you have deployed Docker compose, as explained in the following procedure.

To deploy the containers, docker-compose is required on the local workstation. Perform the following steps:

1. Browse to `/usr/local/bin` and create a file with name `docker-compose` with the following contents:

```
#
# This script will attempt to mirror the host paths by using volumes for the
# following paths:
# * $(pwd)
# * $(dirname $COMPOSE_FILE) if it's set
# * $HOME if it's set
#
# You can add additional volumes (or any docker run options) using
# the $COMPOSE_OPTIONS environment variable.
#

set -e

VERSION="1.27.4"
IMAGE="ibmcom/dockercompose-s390x:$VERSION"

# Setup options for connecting to docker host
if [ -z "$DOCKER_HOST" ]; then
    DOCKER_HOST='unix:///var/run/docker.sock'
fi
if [ -S "${DOCKER_HOST#unix://}" ]; then
    DOCKER_ADDR="-v ${DOCKER_HOST#unix://}:${DOCKER_HOST#unix://} -e DOCKER_HOST"
else
    DOCKER_ADDR="-e DOCKER_HOST -e DOCKER_TLS_VERIFY -e DOCKER_CERT_PATH"
fi

# Setup volume mounts for compose config and context
if [ "$(pwd)" != '/' ]; then
    VOLUMES="-v $(pwd):$(pwd)"
fi
if [ -n "$COMPOSE_FILE" ]; then
    COMPOSE_OPTIONS="$COMPOSE_OPTIONS -e COMPOSE_FILE=$COMPOSE_FILE"
    compose_dir="$(dirname "$COMPOSE_FILE")"
    # canonicalize dir, do not use realpath or readlink -f
    # since they are not available in some systems (e.g. macOS).
    compose_dir="$(cd "$compose_dir" && pwd)"
fi
if [ -n "$COMPOSE_PROJECT_NAME" ]; then
    COMPOSE_OPTIONS="-e COMPOSE_PROJECT_NAME $COMPOSE_OPTIONS"
fi
if [ -n "$compose_dir" ]; then
    VOLUMES="$VOLUMES -v $compose_dir:$compose_dir"
fi
if [ -n "$HOME" ]; then
    VOLUMES="$VOLUMES -v $HOME:$HOME -e HOME" # Pass in HOME to share docker.config and allow
    ~/-relative paths to work.
fi
i=$#
while [ $i -gt 0 ]; do
    arg=$1
```



```

i=$((i - 1))
shift

case "$arg" in
    -f|--file)
        value=$1
        i=$((i - 1))
        shift
        set -- "$@" "$arg" "$value"

        file_dir=$(realpath "$(dirname "$value")")
        VOLUMES="$VOLUMES -v $file_dir:$file_dir"
        ;;
    *) set -- "$@" "$arg" ;;
esac
done

# Setup environment variables for compose config and context
ENV_OPTIONS=$(printenv | sed -E "/^PATH=.*\/d; s\/^\/-e /g; s\/=.*\/g; s\/\n\/ /g")

# Only allocate tty if we detect one
if [ -t 0 ] && [ -t 1 ]; then
    DOCKER_RUN_OPTIONS="$DOCKER_RUN_OPTIONS -t"
fi

# Always set -i to support piped and terminal input in run/exec
DOCKER_RUN_OPTIONS="$DOCKER_RUN_OPTIONS -i"

# Handle userns security
if docker info --format '{{json .SecurityOptions}}' 2>/dev/null | grep -q 'name=userns'; then
    DOCKER_RUN_OPTIONS="$DOCKER_RUN_OPTIONS --userns=host"
fi

# shellcheck disable=SC2086
exec docker run --rm $DOCKER_RUN_OPTIONS $DOCKER_ADDR $COMPOSE_OPTIONS $ENV_OPTIONS $VOLUMES -w "$(pwd)"
$IMAGE "$@"

```

2. Run the following command to make the `docker-compose` file an executable file:

```
sudo chmod +x /usr/local/bin/docker-compose
```

3. More detailed technical information for each component are available in the sample readme files:

- [HCL Workload Automation Server](#)
- [HCL Workload Automation Console](#)
- [HCL Workload Automation dynamic agent](#)
- [HCL Workload Automation z-centric agent](#)
- [Workload Automation FileProxy](#)

Deploying containers with Docker

How to deploy the current version of HCL Workload Automation using Docker containers.

About this task

This chapter describes how to deploy the current version of HCL Workload Automation using Docker containers.

The available Docker containers are:

- HCL Workload Automation Server, containing the master domain manager and backup master domain manager images for UNIX, Windows, and Linux on Z operating systems.
- HCL Workload Automation Console, containing the Dynamic Workload Console image for UNIX, Windows, Linux on Z operating systems, and the IBM z/OS Container Extensions (zCX) feature.
- HCL Workload Automation dynamic agent and the image of the agent with the machine learning engine, containing the Agent image for UNIX, Windows, Linux on Z operating systems. and the IBM z/OS Container Extensions (zCX) feature.
- HCL Workload Automation z-centric agent, containing the Agent image for UNIX, Windows, Linux on Z operating systems. and the IBM z/OS Container Extensions (zCX) feature.

Each container package includes also a `docker-compose.yml` file to configure your installation.

The dynamic agent component (also the one included in the HCL Workload Automation Server container) is deployed and configured with a gateway.

You can choose to deploy all product containers with a single command, or you can deploy each product component container individually.

Deploying all product component containers with a single command

The following readme file contains all the steps required to deploy all product components at the same time:

[HCL Workload Automation](#)

Deploying each product component container individually

If you want to install server, console and agent containers individually, see the related readme files :

- [HCL Workload Automation Server](#)
- [HCL Workload Automation Console](#)
- [HCL Workload Automation dynamic agent](#)
- [HCL Workload Automation z-centric agent](#)
- [Workload Automation FileProxy](#)



Note: The database is always external to the Docker engine and is not available as a container



Note: When deploying the server (master domain manager) container, the database schema is automatically created at the container start. If you are planning to install both the HCL Workload Automation server master domain



manager and backup master domain manager, ensure that you run the command for one component at a time. To avoid database conflicts, start the second component only when the first component has completed successfully.

Accessing the Docker containers

This topic shows you how to access the container shell and run HCL Workload Automation commands.

To check the container status and run HCL Workload Automation commands, you need to access the containers as described below:

1. Obtain the container ID by running the following command: `docker ps`

An output similar to the following one is returned:

CONTAINER ID	IMAGE	NAMES
b02459af2b9c	wa-console

2. Access the Docker container by running the following command: `docker exec -it <container_id> /bin/bash`

Where

container_id

Is the ID of the container obtained with the command explained in the first step, for example

b02459af2b9c.

Connecting an on-prem fault tolerant agent to an HCL Workload Automation Server container

To establish a communication between an on-prem fault tolerant agent and an HCL Workload Automation server container, configure the server *docker-compose.yml* file as follows:

1. Open all external ports as shown in the example below:

```
ports (port mapping "external:internal"):
  - 31116:31116 #HTTPS server port
  - 31111:31111 #HTTP netman port
  - 33113:33113 #HTTPS netman ssl port
  - 31131:31131 #HTTP EIF port
  - 35131:35131 #HTTPS EIF ssl port
```

2. Add the **extra_hosts** parameter under **hostname**, and insert all remote machine hostnames that docker must contact (including the one of the on-prem FTA).

```
hostname: wa-server
extra_hosts:
  - hostname1: IP_Address
  - hostname2: IP_Address
  - hostname3: IP_Address
  ...
```

Furthermore, in the `/etc/hosts` file on the remote machine where the on-prem FTA is running, add the hostname of the HCL Workload Automation server container.

IP_Address	hostname
------------	----------



Note: You can find the hostname of the HCL Workload Automation server container in the server `docker-compose.yml` file.

Creating a Docker image to run dynamic agents

Quickly create a Docker image to run dynamic agents.

You can run dynamic agents in a Docker container that you use to run jobs remotely, for example to call REST APIs or database stored procedures, or to run jobs within the container itself.

To create a Docker container, you are provided with step-by-step instructions and the latest versions of the required samples on GitHub [here](#). Follow the instructions to create a Docker container to run jobs remotely, or use it as base image to add the applications to be run with the agent to other images, or customize the samples to best meet your requirements.

Deploying HCL Workload Automation components on Red Hat OpenShift using helm charts

Deploy the HCL Workload Automation product component containers on a Red Hat OpenShift environment by using helm charts.

The HCL Workload Automation product components can be deployed onto Red Hat OpenShift. You can deploy HCL Workload Automation components using certified containers on a Kubernetes-based container application platform useful to orchestrate containerized applications. You can then manage the HCL Workload Automation containers from the OpenShift dashboard or from the command line interface.

For technical information about the deployment, see [Deploy HCL Workload Automation using helm charts](#).

Deploying from Amazon Web Services (AWS) Marketplace

This publication provides detailed information about how to find, subscribe and deploy HCL Workload Automation directly on AWS Marketplace. A cloud deployment ensures access anytime anywhere and is a fast and efficient way to get up and running quickly.

As more and more organizations move their critical workloads to the cloud, there is an increasing demand for solutions and services that help them easily migrate and manage their cloud environment. To respond to the growing request to make automation opportunities more accessible, HCL Workload Automation can be deployed from AWS Marketplace.

In the following topics, you can find the procedures to subscribe, deploy and upgrade HCL Workload Automation from AWS Marketplace:

1. [Getting started on page 177](#)
2. [Creating stacks on AWS CloudFormation on page 178](#)

3. [Accessing the cluster environment and getting credentials on page 179](#)
4. [Downloading packages from the Dynamic Workload Console on page 182](#)
5. [Integrating AIDA on page 182](#)
6. [Upgrading on page 183](#)
7. [Uninstalling on page 184](#)

Meter usage

By deploying HCL Workload Automation on AWS, you receive the bill directly on your AWS account. Through the AWS Marketplace Metering Service, the amount of the bill varies according to the number of jobs that are submitted in your HCL Workload Automation environment. For more information about AWS billing management, see [AWS Billing and Cost Management Documentation](#).

Related information

[Getting started on page 177](#)

Getting started

You can deploy HCL Workload Automation quickly and easily on Amazon Web Services (AWS) using AWS CloudFormation resources.

About this task

This topic describes the main steps to complete before deploying the product, such as subscribing to HCL Workload Automation on AWS Marketplace.

You can find the procedure to subscribe to HCL Workload Automation on AWS below.

1. Go to [AWS Marketplace](#) and log in.
2. In the home page, click on **View all products**.
3. In the **Search** field, type **HCL Workload Automation** and press Enter.
4. In the result page, select **HCL Workload Automation**.
5. In the product page, click **Continue to subscribe** and proceed with the subscription.

Results

You have now subscribed to HCL Workload Automation on AWS Marketplace.

What to do next

To proceed with the deployment of HCL Workload Automation, you have to create stacks on AWS CloudFormation.

Related information

[Creating stacks on AWS CloudFormation on page 178](#)

Creating stacks on AWS CloudFormation

In this topic you can find the prerequisites and the steps to create stacks on AWS CloudFormation.

Before you begin

After having [subscribed to HCL Workload Automation on AWS Marketplace on page 177](#), you can start the stack creation process on AWS CloudFormation.

To deploy HCL Workload Automation, your environment needs to meet the following prerequisites:

- An *IAM* identity, whether *role* or *user*, with admin permission on your AWS account.
- Helm 3.0
- `kubectl` command-line tool to control Kubernetes clusters.
- AWS Command Line Interface (AWS CLI) tool.



Note: If you have an existing cluster, make sure that you have AWS admin roles to deploy Kubernetes resources and HCL Workload Automation Helm Chart resources.

Resources required

The following resources correspond to the default values required to manage a production environment. These numbers might vary depending on the environment.

Component	Container resource limit	Container memory request
Server	CPU: 4, Memory: 16Gi	CPU: 1, Memory: 6Gi, Storage: 10Gi
Console	CPU: 4, Memory: 16Gi	CPU: 1, Memory: 4Gi, Storage: 5Gi
Dynamic Agent	CPU: 1, Memory: 2Gi	CPU: 200m, Memory: 200Mi, Storage size: 2Gi
File Proxy	CPU: 100m, Memory: 128Mi	CPU: 100m, Memory: 128Mi

About this task

You can create stacks on AWS CloudFormation using two templates. The first template deploys the EKS cluster, while the second template deploys the product prerequisites, the resources and the HCL Workload Automation Helm Chart.

Follow the procedure for each template.



Note: If you already have an existing cluster that meets the prerequisites, you can follow the procedure for the second template only.

1. Go to [Amazon Web Services](#) and log in.



Note: By default, the time zone on the AWS CloudFormation home page is set on **N. Virginia**. You can edit the time zone by selecting the time zone that you want from the menu on the right side of the home page.

2. Select the template corresponding to the stack you want to create.

Deploy EKS cluster

Deploy HWA

3. Click **Next**.
4. In the **Specify stack details** page, type a stack name and complete all the fields with the required information.
5. Click **Next**.
6. In the **Configure stack options** page, in the **Permissions** sections type the *IAM role* or *IAM user* name.
7. Click **Next**.
8. In the **Review** page, review the details of the stack.
9. Click **Submit**.

Results

AWS CloudFormation constructs and configures the stack resources specified in the templates.

What to do next

Next, you have to access the environment and get the user credentials.

Related information

[Accessing the cluster environment and getting credentials on page 179](#)

Accessing the cluster environment and getting credentials

In this topic you can find information about how to access your HCL Workload Automation environment and how to get user credentials.

After having created the stacks on AWS CloudFormation, you can access the environment and get the credentials to log in. When you get your credentials, you can validate the deployment by verifying the installation manually.

Accessing the cluster environment

To access HCL Workload Automation environment, you can follow the procedure described on Amazon EKS User Guide. See [Creating or updating a `kubeconfig` file for an Amazon EKS cluster](#).

Getting credentials

After having accessed the environment, you need to get the credentials to access to the Dynamic Workload Console. You can find the steps to get the credentials below.

For load balancer:

1. Run the following command to obtain the token to be inserted in `https://<loadbalancer>:9443/console` to connect to the console:

```
kubectrl get svc <workload_automation_release_name>-waconsole-lb -o  
'jsonpath={..status.loadBalancer.ingress..hostname}' -n <workload_automation_namespace>
```

2. With the output obtained, replace <loadbalancer> in the URL `https://<loadbalancer>:9443/console`.

For ingress:

1. Run the following command to obtain the token to be inserted in `https://<ingress>/console` to connect to the console:

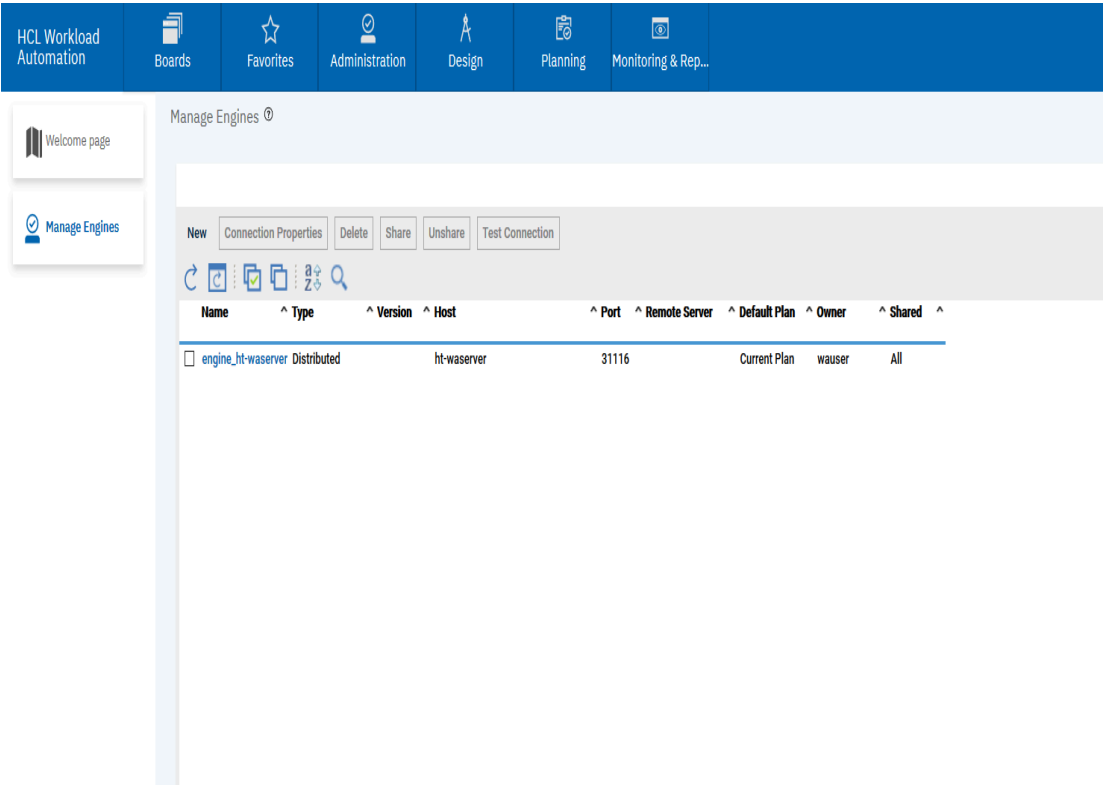
```
kubectrl get ingress/<workload_automation_release_name>-waconsole -o 'jsonpath={..host}' -n  
<workload_automation_namespace>
```

2. With the output obtained, replace <ingress> in the URL `https://<ingress>/console`.

Logging into the console:

1. Log in to the console by using the URLs obtained in the previous step.
2. For the credentials, specify the user name (wauser) and the password specified in the "HWA Console and Server Password" parameter of the prerequisites, resources and HCL Workload Automation Helm Chart deployment template.

- 3. From the navigation toolbar, select **Administration -> Manage Engines**.
- 4. Verify that the default engine, **engine_<release>-waserver** is displayed in the Manage Engines list



release:

To ensure the Dynamic Workload Console logout page redirects to the login page, modify the value of the logout url entry available in file authentication_config.xml:

```
<jndiEntry value="${logout.url}" jndiName="logout.url" />
```

where the logout.url string in jndiName should be replaced with the logout URL of the provider.

Verifying the installation

When the deployment procedure has completed, you can validate the deployment to ensure that everything is working. To verify the successfulness of the installation, you can perform the checks described [here](#).

What to do next

Next, you can integrate the agent or the Orchestration CLI in your environment. For more information, see [Downloading packages from the Dynamic Workload Console on page 182](#).

Related information

[Downloading packages from the Dynamic Workload Console on page 182](#)

Downloading packages from the Dynamic Workload Console

To integrate HCL Workload Automation agent and Orchestration CLI in your environment, you can download the packages from the Dynamic Workload Console.

About this task

After the deployment of HCL Workload Automation, you can integrate agent and Orchestration CLI in your environment.

The following scenario describes the steps to download the packages from the Dynamic Workload Console and to extract the executable file.

Procedure:

1. Log in as **Administrator** to the Dynamic Workload Console.
2. From the **Admin** menu, click **Download Center** in the **Download** section.
3. In the **Download Center** page, choose the package you want to download.
4. Click on the card corresponding to your operating system and wait for the download of the package to start.
5. Extract the executable file from the downloaded package and save it on any path that you want.

Results

You have now downloaded and extracted the package and you can start the installation or configuration process.

For more information about agent installation, see: [Installing agents on page 113](#).

For more information about Orchestration CLI configuration, see: [Configuring Orchestration CLI](#).

What to do next

Next, you can integrate AIDA in your environment.

Related information

[Integrating AI Data Advisor \(AIDA\) on page 182](#)

Integrating AI Data Advisor (AIDA)

In this topic you can find information about how to integrate AIDA in your environment.

AI Data Advisor (AIDA) is a component delivered starting with HCL Workload Automation V 10.1, based on Artificial Intelligence and Machine Learning techniques. It enables fast and simplified data-driven decision making for an intelligent workload management.

To configure AIDA, run the following commands:

1. Export the HELM_EXPERIMENTAL_OCI variable

```
export HELM_EXPERIMENTAL_OCI=1
```

2. Authenticate to your AWS account.

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin
*****.dkr.ecr.us-east-1.amazonaws.com
```

3. Pull the chart.

```
helm pull oci://*****.dkr.ecr.us-east-1.amazonaws.com/hcl-technologies/hcl-workload-automation-prod
```

4. Get the current deployment values.

```
helm get values <RELEASE_NAME>
```

5. Open the values.yaml file and set the enableAIDA parameter to true:

```
enableAIDA: true
```

For more information, see [AIDA Configuration Parameters](#).

You can find the AIDA readme file at the following link: [AI Data Advisor \(AIDA\)](#).

Related information

[Upgrading on page 183](#)

Upgrading

After completing the deployment process of HCL Workload Automation, you can upgrade or customize the chart.

In this topic you can find information about how to upgrade your HCL Workload Automation environment, either to a new version of the product or with new HCL Workload Automation components.

To upgrade the release <workload_automation_release_name> to a new version of the chart, run the following commands:

1. Export the HELM_EXPERIMENTAL_OCI variable

```
export HELM_EXPERIMENTAL_OCI=1
```

2. Authenticate to your AWS account.

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin
*****.dkr.ecr.us-east-1.amazonaws.com
```

3. Pull the chart.

```
helm pull oci://*****.dkr.ecr.us-east-1.amazonaws.com/hcl-technologies/hcl-workload-automation-prod
```

4. Get the current deployment values.

```
helm get values <RELEASE_NAME>
```

The output of the helm get values command is file named values.yaml. You can customize the values of this file according to the configuration parameters listed [here](#).

5. Upgrade the chart.

```
helm upgrade
  release_name //*****.dkr.ecr.us-east-1.amazonaws.com/hcl-technologies/hcl-workload-automation-prod -f
  values.yaml -n <workload_automation_namespace>
```

For more information about this command, see: [Upgrading the Chart](#)

Uninstalling

In this topic you can find information about how to uninstall the chart, clean up the orphaned Persistent Volumes and deprovision AWS resources.

To uninstall the deployed components associated with the chart and clean up the orphaned Persistent Volumes, run the following commands:

1. Uninstall the hcl-workload-automation-prod deployment.

```
helm uninstall release_name -n <workload_automation_namespace>
```

The command removes all of the Kubernetes components associated with the chart and uninstalls the <workload_automation_release_name> release.

2. Clean up orphaned Persistent Volumes.

```
kubectl delete pvc -l <workload_automation_release_name> -n <workload_automation_namespace>
```

3. Proceed with the deprovisioning of the deployed resources by deleting the stack on AWS CloudFormation. For more information, see [Deleting a stack on the AWS CloudFormation console](#).
4. Check if all the deployed resources, such as NGINX Ingress Controller, cert-manager, HCL Workload Automation and their related namespaces, have been deleted from the cluster. If not, run the following command to delete them manually:

```
kubectl delete ns <namespace>
```

When you delete the namespace, also the related resources inside that namespace are deleted.

Deploying on Amazon EKS

You can use Amazon Elastic Kubernetes Service (EKS) to run HCL Workload Automation containers on Amazon Web Services (AWS) EKS.

As more and more organizations move their critical workloads to the cloud, there is an increasing demand for solutions and services that help them easily migrate and manage their cloud environment.

To respond to the growing request to make automation opportunities more accessible, HCL Workload Automation is now offered on the Amazon Web Services cloud. Within just a few minutes, you can access the product Helm chart and container images and easily launch an instance to deploy an HCL Workload Automation server, console, and agents with full on-premises capabilities on AWS. HCL Workload Automation on AWS improves flexibility and scalability of your automation environment. It helps in lowering costs and eliminating complexity, while reducing the operational overhead and the burden involved in managing your own infrastructure, so you can invest your time and resources in growing your business. Also, HCL Workload Automation on AWS delivers faster access to managed services solutions, for a full product lifecycle management.

Full details and deployment instructions are available in the [HCL Workload Automation Chart readme file](#). You can also subscribe to HCL Workload Automation on EKS on [AWS Marketplace](#).

Deploying on Azure AKS

You can deploy and manage HCL Workload Automation containers on Azure Kubernetes Service (AKS).

Deploy and manage HCL Workload Automation containerized product components on the Azure AKS, a container orchestration service available on the Microsoft Azure public cloud. You can use Azure AKS to deploy, scale up, scale down and manage containers in the cluster environment. You can also deploy and run an Azure SQL database.

As more and more organizations move their critical workloads to the cloud, there is an increasing demand for solutions and services that help them easily migrate and manage their cloud environment.

To respond to the growing request to make automation opportunities more accessible, HCL Workload Automation can now be deployed on Azure AKS. Within just a few minutes, you can easily launch an instance to deploy an HCL Workload Automation server, console, and agents with full on-premises capabilities on the Microsoft Azure public cloud.

HCL Workload Automation deployed in a cluster environment improves flexibility and scalability of your automation environment. It helps in lowering costs and eliminating complexity, while reducing the operational overhead and the burden involved in managing your own infrastructure, so you can invest your time and resources in growing your business.

Running the product containers within Azure AKS gives you access to services such as a highly scalable cloud database service. You can deploy and run any of the following Azure SQL Server database models in the Azure cloud, depending on your needs:

- SQL database
- SQL managed instance
- SQL virtual machine

Full details and deployment instructions are available in the [HCL Workload Automation Chart readme file](#).

Deploying on Google GKE

You can deploy and manage HCL Workload Automation containers on Google GKE.

Google Kubernetes Engine (GKE) provides a managed environment for deploying, managing, and scaling your containerized applications using Google infrastructure. The Google GKE environment consists of multiple machines grouped together to form a cluster. You can also deploy and run Google Cloud SQL for SQL server.

As more and more organizations move their critical workloads to the cloud, there is an increasing demand for solutions and services that help them easily migrate and manage their cloud environment.

To respond to the growing request to make automation opportunities more accessible, HCL Workload Automation can now be deployed on Google GKE. Within just a few minutes, you can easily launch an instance to deploy an HCL Workload Automation server, console, and agents with full on-premises capabilities on the Google GKE public cloud.

HCL Workload Automation deployed in a cluster environment improves flexibility and scalability of your automation environment. It helps in lowering costs and eliminating complexity, while reducing the operational overhead and the burden involved in managing your own infrastructure, so you can invest your time and resources in growing your business.

Running the product containers within Google GKE gives you access to services, such as a cloud database service. Cloud SQL for SQL Server is a managed database service that helps you set up, maintain, manage, and administer your SQL Server databases on Google Cloud Platform

Full details and deployment instructions are available in the [HCL Workload Automation Chart readme file](#).

Workload Automation on HCL SoFy

An easy way to deploy HCL Workload Automation product components as Docker containers to a cloud-native environment.

HCL SoFy includes all the tools required to build a Kubernetes deployment package for HCL Workload Automation, and run it on the Kubernetes cloud of your choice (public or private). SoFy solutions are portable across all Kubernetes environments. You can also take advantage of SoFy's unique Kubernetes monitoring and administration tools.

HCL SoFy uses Helm technology to provide HCL products and application programming interfaces. A temporary sandbox environment is provided to deploy and test solutions. You can run more than two temporary sandboxes at the same time.

Accessing a Dynamic Workload Console from a SoFy solution

You can also access an HCL Workload Automation Dynamic Workload Console from a SoFy following these steps:

1. Go to <https://hclsofy.com>
2. Access the **Sandboxes** page
3. After accessing the **Sandboxes** page, you get access to the following **sandbox links**:
 - Documentation site
 - Service Now
 - Your Automation Hub
 - HCL Workload Automation Console

Select **HCL Workload Automation Console** to access the Dynamic Workload Console



Note: In **sandbox links**, in the **HCL Workload Automation Console** section, the SoFy solution automatically generates a login ID and a password, which you can use to access the **SoFy Console**, where you can monitor all the pods currently deployed on the environment.

In the **HCL Workload Automation Console** section, the SoFy solution automatically generates a login ID and a password, that you can use to access the **SoFy Console**, where you can monitor all the pods currently deployed on the environment.

For complete details about deploying services and products enabled for Kubernetes as docker images and helm charts, see the [HCL SoFy Guides](#). Search for HCL Workload Automation in the HCL SoFy Catalog here <https://hclsofy.com>.

HCL Workload Automation on Now Readme File

This section provides important information about HCL Now, HCL's Cloud-Native-as-a-Service that allows the freedom to deploy anywhere, power to innovate, and flexibility to scale on a dedicated environment.

To try out the HCL Workload Automation solution, you can request access to HCL SoFy, HCL's Solution Factory where you can get started deploying HCL Workload Automation in Kubernetes through cloud-centric technologies and practices. HCL Workload Automation on Now is available on the same HCL SoFy portal. For more information about HCL SoFy see [HCL SoFy](#).

This readme file is the most current information for HCL Workload Automation on Now and takes precedence over all other documentation for this offering.

It is divided into the following sections:

- [About HCL Workload Automation on Now on page 187](#)
- [Known limitations and workarounds on page 187](#)
- [Installing the agent on page 188](#)
- [Reference on page 190](#)

For the most up-to-date information about supported operating systems for the dynamic agent, see the **Supported Software** document available at Product Requirements.

About HCL Workload Automation on Now

HCL Workload Automation on Now offers a fully managed cloud experience. Outsource the administration and management of the server to HCL Software so you can concentrate on automating and enhancing your business processes.

HCL Now managed services take care of security and compliance and ongoing maintenance and upgrades to the latest releases of HCL Workload Automation.

HCL Workload Automation on Now uses the latest agent technology (dynamic agents) so that you have access to all of the job types available. The dynamic agent also complies with the latest security guidelines and requirements. Install dynamic agents where you want to define, run, and monitor your workloads, on cloud or on-premises.

After installing your agents, you model, submit, and monitor your workload using the Dynamic Workload Console. For more information about these operations, see the links in the section [Reference on page 190](#).

Known limitations and workarounds

The following are software limitations that affect HCL Workload Automation on Now.

Security

Access control lists are used to assign security roles to users or groups on a specific folder. Security domains are not supported.

Command-line interface

The composer and conman command-line interfaces are not supported.

AI Data Advisor (AIDA)

AIDA is not supported on HCL Now.

FINAL job stream

The FINAL job stream is managed exclusively by HCL Software. The FINAL job stream runs the sequence of script files described in JnextPlan to generate the new production plan.

The following is a workaround you can apply when working with HCL Workload Automation on Now:

If you need to retrieve the password and have access to the **sofy-console**, you can browse to the following section: **Solution Content -> HCL Workload Automation -> General Information -> API Links**.

Installing the agent

Install agents where you want to define, run, and monitor your workloads.

Ensure that the user credentials used to install the agent have full control or, at a minimum, display privileges on the file AGENT_CERTIFICATE. For more information about how to create a user, see [Downloading certificates or JWT using a different user on page 487](#).

To install the agent, follow these high-level steps:

On Windows™ operating systems:

1. Download the agent software image from [HCL Software](#) to your local workstation. Ensure you have enough temporary space.
2. Log in as administrator on the workstation where you want to install the agent.
3. From the *image_directory\TWS\operating_system* directory, run `twsinst` by using the following syntax. Only the minimum set of parameters necessary to install the dynamic agent are specified.

```
cscript twsinst.vbs -new
  -uname user_name
  -password user_password
  -acceptlicense yes
  -agent dynamic
  -displayname display_name
  -inst_dir installation_directory
  -jimport 31114
  -tdwbport 31116
  -tdwbhostname your_hwa_instancename.hxwa.now.hclsoftware.cloud
  -wouser wouser
  -wapassword wa_password
  -gateway local
  -gwid gateway_id
```

For a description of the syntax parameters and a complete list of them, see [Agent installation parameters - twsinst script on page 119](#).



Note: `twinsinst` for Windows™ is a Visual Basic Script (VBS) that you can run in CScript and WScript mode.

The HCL Workload Automation user is automatically created. The software is installed by default in the HCL Workload Automation installation directory. The default value is `%ProgramFiles%\HCL\TWA`.

If you enabled the Security Warning, a dialog box is displayed during the installation. In this case, answer Run to continue.

On UNIX™ and Linux™ operating systems:

1. Download the agent software image from [HCL Software](#) to your local workstation. Ensure you have enough temporary space.
2. If you plan to log in as **root** on the workstation where you will install the agent, create the HCL Workload Automation user. The software is installed by default in the user's home directory, referred to as `/installation_dir/TWS`.

User:

`TWS_user`

Home:

`/installation_dir/TWS` (for example: `/home/user1/TWS` where `user1` is the name of HCL Workload Automation user). Ensure this directory has **755** permission.

If you plan to log in as a non-root user (available only for dynamic agents), your login will become by default the only possible user of the agent. You do not need to create another HCL Workload Automation user, but make sure that you have a home directory (where the agent will be installed), and that it has **755** permission.



Important: If you use the `-su non-root username` command in the shell where you are about to run `twinsinst`, make sure that `$HOME` is set on your home directory as a non-root user (use `echo $HOME` to verify that the value returned corresponds to your home directory).

3. Log in on the workstation where you want to install the agent.
4. From the `image_directory/TWS/operating_system` directory, run `twinsinst` by using the following syntax. Only the minimum set of parameters necessary to install the dynamic agent are specified.

```
./twinsinst -new
  -uname user_name
  -acceptlicense yes
  -agent dynamic
  -displayname display_name
  -inst_dir installation_directory
  -reset_perm
  -skipcheckprereq
  -tdwbpport 31116
```

```
-tdwbhostname your_hwa_instancename.hxwa.now.hclsoftware.cloud
-wauser wauser
-wapassword wa_password
-gateway local
-gwid gateway_id
```

For a description of the syntax parameters, see [Agent installation parameters - twsinst script on page 119](#)

If the installation fails, to understand the cause of the error see [Analyzing return codes for agent installation, upgrade, restore, and uninstallation on page 400](#).

Reference

Refer to the following links for information about frequent tasks you perform using HCL Workload Automation on Now:

Configuring the dynamic agent

[Configuring a dynamic agent on page 216](#)

Navigating the Dynamic Workload Console

[Navigating the Dynamic Workload Console](#)

HCL Workload Automation concepts

[HCL Workload Automation Concepts](#)

Modeling your workload:

[Designing your workload](#)

Managing workload security

[Managing Workload Security](#)

Submitting the workload to run

[Submitting Workload on Request in Production](#)

Monitoring your workload

[Monitoring your environment](#)

Simulating changes to analyze the impact with the What-if analysis Gantt chart

[Analyzing the impact of changes on your environment](#)

Deploying AI Data Advisor

You can deploy AI Data Advisor (AIDA) by using Docker or Kubernetes.

AIDA is composed of two major components: AIDA Exporter and AIDA Engine. Each component contains a number of services:

AIDA Exporter**Exporter**

Through HCL Workload Automation APIs, exports KPIs metrics from HCL Workload Automation (according to OpenMetrics standard) and stores them into AIDA OpenSearch database.

Also, it exports Alert definitions from HCL Workload Automation and imports them into OpenSearch.

AIDA Engine**Predictor**

Calculates the expected values of each KPI, also considering special days.

Anomaly detection and alert generation

Detects anomalies in KPIs trend by comparing observed KPI data points with expected values, and generates alerts when trigger conditions are met.

Email notification

Sends email notification when alerts are generated.

Orchestrator

Orchestrates KPI prediction and anomaly detection.

UI

AIDA User Interface.

Internal event manager

Manages communication among AIDA services.

Also, AIDA uses:

OpenSearch (an Elasticsearch technology)

To store and analyze data.

Keycloak

To manage security and user access in AIDA (Docker deployment only). If not deployed, the Dynamic Workload Console user authentication roles will be used.

Nginx

As a reverse proxy for its components.

Deploying AIDA using Docker

To monitor HCL Workload Automation and HCL Workload Automation for Z engines, you can deploy AIDA with Docker by using AIDA.sh script. This script provides options to run Docker Compose operations and AIDA configuration steps.

For details, see the following readme file for Docker:

- [HCL AI Data Advisor for HCL Workload Automation](#)

Deploying AIDA using Kubernetes

To monitor HCL Workload Automation engines only (not HCL Workload Automation for Z engines), you can deploy AIDA by using an **helm chart**. This helm chart is included in the Workload Automation product helm chart that allows you to deploy Workload Automation and all its components in one shot.

For details, see the following readme file for Kubernetes:

- [HCL AI Data Advisor for HCL Workload Automation](#)



Note: Horizontal pod autoscaling based on memory and network usage is supported for **Anomaly detection and alert generation** and **Predictor** services. In case of high memory utilization, Kubernetes replicates pods. When memory usage decreases, pods are deleted.

Deploying AIDA on Amazon Web Services (AWS) Marketplace

You can use Amazon Web Services (AWS) Marketplace to subscribe to HCL Workload Automation and deploy your environment on the AWS secure cloud platform, including AIDA as optional component.

For more information see [Deploying from Amazon Web Services \(AWS\) Marketplace on page 176](#).

Troubleshooting

This section describes how to resolve problems with HCL Workload Automation containers. It describes the tools available to help you troubleshoot problems and details many known problem scenarios, and their solutions.

Container maintenance procedure

Check how to avoid POD restart during maintenance.

The POD status check of a Kubernetes-based environment is based on Liveness Probe; the latter checks if all processes are active, if one or more processes are not active, the POD is automatically restarted. Therefore, in case of maintenance, manually stopping the HCL Workload Automation processes, the Dynamic Workload Console processes, or the dynamic agent processes causes a POD restart.

To avoid a PD restart during maintenance:

- In the selected POD, create the following file: `/opt/wautils/wa_maintenance`
- Stop the processes listed above and perform all the steps needed for the maintenance
- Restart the stopped processes
- Delete the `/opt/wautils/wa_maintenance` file

Container deployment issues

Check the steps to do if you run into deployment issues.

About this task

If a problem occurs during the deployment, check the steps described below to solve it.

Docker compose

1. Check the system requirements [Prerequisite information when deploying with containers on page 171](#)
2. Make sure that all required configuration parameters have been correctly configured (e.g. license, WA_PASSWORD, DB parameters, etc.).
3. Make sure that the external port mapping does not collide with ports already used by other processes.
4. Activate the debug mode by performing the following steps:
 - Remove all containers by launching the "docker rm -f wa-server wa-console wa-db2" command.
 - Remove the associated volumes by launching the "docker volume prune" command.
 - Edit the docker-compose.yml file by adding "- WA_DEBUG=yes?" under the environment variables (this prevents the containers to exit after the failure).
 - Launch again the services by using "docker-compose up -d".
 - Enter the container name by using "docker exec -it *container_name_or_id* /bin/bash".
 - Check the logs
5. If you find an error in the logs, check the detailed logs in /home/wauser/wadata/installation/logs

Red Hat OpenShift

1. See the system requirements documented in the readme [IBM Workload Automation](#) for OpenShift V4.x.
2. Make sure that all required configuration parameters have been correctly configured in the custom resources (OpenShift 4.x) or in the *template.yml* file (OpenShift 3.x), for example, license, pools, storage class, to name a few.
3. From the OpenShift command line, check the POD logs by launching the "oc logs -f *pod_name*" command. If launched with the -f (--follow) option, it shows useful information about the installation phase. From the OpenShift platform, go to *Stateful Sets* section in *Applications*, double click on PODS and then click on the POD's name to see the related logs.
4. Activate the debug mode to check the container installation and configuration logs by setting true the "WA_DEBUG" parameter in the configuration file.
5. If you find an error in the logs, check the detailed logs in /home/wauser/wadata/installation/logs

"CURL error 35" error

This document explains how to solve the *CURL error 35* error that might occur on the agent.

If in the *JobManagerGW_message.log* file on the agent you find the following error:

```
|18446744072657463040|152|agent-95-waagent-0.agent-95-waagent-h.cert-manager.svc.cluster.local|
AWSITA320E The gateway was not able to contact the broker server at the address
"https://localhost:35116/JobManagerRESTWeb/JobSchedulerGW/actions/GWID_AGENT_ICP_agent_95_waagent_0"
to obtain the list of actions to execute.
The error is: "AWSITA245E An error occurred getting the response of the HTTP request.
The error is "CURL error 35".
```

And simultaneously in the *message.log* on the server you find the following error:

```
00058543 com.ibm.scheduling.jobdispatcher
W AWKJDE235W A connection problem occurred submitting job ID "25f769bd-d0e3-3a90-ae47-c7f8a51c549c" with name
"AGENT_ICP#EVERY_1800_4.S_PEAK_JOB_65.SCHEDID-0AAAAAAAAAAP35AZ.JNUM-757735705" to the endpoint URL
"https://agent-95-waagent-0:31114/ita/JobManager/job".
The error message is: "AWKJDE519E The agent did not contact the server to manage this request.".
```

Proceed as follows:

1. Edit the *JobManagerGW.ini* file on the agent, by adding **ActionPollers = 3** (if the ActionPollers is not specified, the default value is 1). The file is located in the following path:

```
/home/wauser/wadata/ITA/cpa/config/
```



Note: The **ActionPollers = 3** must be added only in the *[ITA]* section.

The following is an example of the *JobManagerGW.ini* file:

```
[ITA]
name = JobManagerGW
autostart = yes
fname = /opt/wa/TWS/bin/JobManagerGW
keepalive = yes
status_timeout = 300
check_status = yes
commstart = false
display_name = JobManagerGW
version = 1.0
type = optional
min_up_time = 60
JobManagerGWID = GWID_AGENT_ICP_agent_95_waagent_0
JobManagerGWURIs = https://localhost:31114/ita/JobManagerGW/JobManagerRESTWeb/
                   JobScheduler/resource
ActionPollers = 3
```

2. To avoid a POD restart during maintenance, follow the procedure described in [Container maintenance procedure on page 192](#).
3. Stop and start the agent by submitting the following command:

```
/opt/wa/TWS/ShutDownLwa
```

```
/opt/wa/TWS/StartUpLwa
```

Chapter 7. Post-installation configuration

The most common configuration steps to be performed after completing the installation.

After successfully installing HCL Workload Automation, there are a number of recommended configuration steps to be performed that are described in more detail in this section. Also consider the FAQs listed below.

FAQ - Security configurations

A list of questions and answers related to security configurations:

When installing the HCL Workload Automation, you might have the need to customize some parameters to suit your environment.

Can I install any HCL Workload Automation components without setting up the SSL configuration?

No, starting from version 10.2.1, certificates, either default or custom, are required when installing HCL Workload Automation. You can no longer install HCL Workload Automation without securing your environment with certificates.

How do I configure master domain manager and dynamic domain manager in SSL mode?

See the detailed explanation in [Configuring your master domain manager and dynamic domain manager in SSL mode on page 203](#).

How do I configure 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. For more information, see the topic about [Configuring the Dynamic Workload Console for Single Sign-On in *Administration Guide*](#).

Configuring a user registry

In this topic you can find information about how to configure a user registry.

About this task

By default, the dynamic domain manager, the Dynamic Workload Console, and the master domain manager are configured to use a local file-based user repository. For more information about supported authentication mechanisms, see the topic about available configurations in the *Administration Guide*.

You can implement an OpenID Connect (OIDC) user registry, a Lightweight Directory Access Protocol (LDAP) user registry, or a basic user registry by configuring the sample authentication templates provided in XML format. 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 related Open Liberty documentation, for example [LDAP User Registry \(IdapRegistry\)](#).

To configure an OIDC user registry, see [Configuring an OIDC user registry on page 196](#).

To configure an LDAP user registry, for example as Active Directory, see [Configuring an LDAP user registry on page 197](#).

To configure a basic user registry, see [Configuring a basic user registry on page 198](#).

Configuring an OIDC user registry

About this task

You can implement an OIDC user registry by configuring the sample authentication template provided in XML format: `openid_connect.xml`.

To configure an OIDC user registry, complete the following steps:

1. Copy the following template to a working directory:

```
<server>
  <featureManager>
    <feature>openidConnectClient-1.0</feature>
  </featureManager>

  <authFilter id="restFilterOpenID">
    <requestUrl id="restUrl" urlPattern="jwt/ibm/api|/dwc/rest/roles|/dwc/ServiceDispatcherServlet?ServiceName=PrefExport|/metrics" matchType="notContain"/>
  </authFilter>

  <openidConnectClient id="keycloak"
    clientId="wa-service"
    clientSecret="put_oidc_secret_here"
    httpsRequired="true"
    userIdentifier="preferred_username"
    signatureAlgorithm="RS256"
    scope="openid"
    authFilterRef="restFilterOpenID"
    inboundPropagation="supported"
    groupIdentifier="groups"
    accessTokenAttributeName="groups"
    groupNameAttribute="groups"
    hostNameVerificationEnabled="false"
    realmName = "your_realm_name"
    redirectToRPHostAndPort="https://dwc_ingress_hostname"
    discoveryEndpointUrl="https://oidc_ingress_hostname/realms/wa/.well-known/openid-configuration">
  </openidConnectClient>
</server>
```

2. Edit the template file in the working folder with the desired configuration by adding users and groups as necessary.
3. Optionally, create a backup copy of the configuration file in the `overrides` folder, if already present.
4. Copy the updated template file to the `overrides` folder.
5. To upload the certificates of the OIDC provider, browse to `<DWC_home>/java/jre/bin` and run the following command:

```
keytool -importcert -file ingress-cert.pem
-keystore <DWC_home>/usr/servers/dwcServer/resources/security/TWSServerTrustFile.p12 -alias
ingress-cert -storepass <password_keystore>
```


where

ingress-cert.pem

The certificates file to be imported into the Dynamic Workload Console.

ingress-cert

The alias defined during the import of the certificate.

6. On Keycloak, ensure you define the group with the same name present in the OpenID file.



Note: If one or more messages similar to the following are displayed, perform the steps listed below.

```
CWPKEI0819I: The default keystore is not created because a password is not configured on the
<keyStore id="defaultKeyStore"/> element, and the 'keystore_password' environment variable is not set.
CWOAU0073E: An authentication error occurred. Try closing the web browser and authenticating again,
or contact the site administrator if the problem persists.
```

1. On the workstation where the Dynamic Workload Console is installed, browse to the `server.xml` file located in `<dwc_installation_directory>/usr/servers/dwcServer`.
2. Open the file with a text editor and change the value of the **sameSiteCookie** parameter from `strict` to `lax`.
3. Optionally, trust the Dynamic Workload Console certificate with the keycloak certificate.

Configuring an LDAP user registry

About this task

You can implement an LDAP based user repository by configuring the following sample authentication templates provided in XML format. The following are the supported authentication methods and the corresponding sample template that can be configured to replace the configuration file currently in use:

- OpenLDAP: `auth_OpenLDAP_config.xml`
- IBM® Directory Server: `auth_IDS_config.xml`
- Windows Server Active Directory: `auth_AD_config.xml`

To configure a common authentication provider for both the HCL Workload Automation and the Dynamic Workload Console, complete the following steps:

1. Assign a role to your authentication provider user or group.
 - a. Log in to the Dynamic Workload Console as administrator and access the **Manage Roles** page.
 - b. Add a new **Entity** of type **Group** to the role you want to assign to your authentication provider user or group and click **Save**.
2. Update the authentication configuration template file with the details about your authentication provider server.
 - a. Copy the template file to a working directory. The templates are located in the following path:

Dynamic Workload Console

```
DWC_DATA_dir/usr/servers/dwcServer/configDropins/templates/authentication
```

master domain manager

```
TWA_DATA_DIR/usr/servers/engineServer/configDropins/templates/authentication
```

Dynamic Workload Console

```
DWC_home\usr\servers\dwcServer\configDropins\templates\authentication
```

master domain manager

```
TWA_home\usr\servers\engineServer\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 in a different directory, if the file is already present. To avoid conflicts, ensure the backup copy is in a directory different from the following directories: `configDropins/templates` and `configDropins/overrides`.
- d. Copy the updated template file to the `overrides` directory.
- e. The `overrides` directory is located in 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
```

- f. Stop and restart Open Liberty using the `stopappserver` and `startappserver` commands located in `TWA_home/appservertools`.

For more information about configuring an LDAP registry, see the Open Liberty documentation, for example: [Configure an LDAP user registry](#) and [Federated user registries](#).

Configuring a basic user registry

About this task

You might want to use a basic user registry by defining the users and groups information for authentication on Open Liberty, even though this type of authentication is not recommended. This type of authentication cannot be used for production, but only for test purposes.

To configure basic user registry, complete the following steps:

1. Copy the `auth_basicRegistry_config.xml` template from the `templates` folder to a working folder.
2. Edit the template file in the working folder with the desired configuration by adding users and groups as necessary.

To add a user, add an entry similar to the following in the **basicRegistry** section:

```
<user name="nonadminuser" password="{xor}Ozo5PiozKw==" />
```

To add a group, add an entry similar to the following in the **basicRegistry** section:

```
<group name="TWSUsers">
  <member name="nonadminuser" />
</group>
```

3. Store the password in xor, aes, or hash formats using the Open Liberty `securityUtility` command, as described in [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:

HCL Workload Automation

```
<INST_DIR>/TWS/JavaExt/jre/jre
```

Dynamic Workload Console

```
<DWC_INST_DIR>/java/jre/bin
```

4. Create a backup copy of the configuration file in the `overrides` folder, if already present.
5. Copy the updated template file to the `overrides` folder. Maintaining the original folder structure is not required.

Open Liberty configuration

Describes how Open Liberty configuration files are organized in HCL Workload Automation

To simplify administration, configuration, and backup and recovery on UNIX systems, a new default behavior has been implemented with regard to the storage of product data and data generated by HCL Workload Automation, such as logs and configuration information. These files are now stored by default in the `TWA_DATA_DIR` directory, which you can optionally customize at installation time.

With a similar approach, also the configuration files for Open Liberty on UNIX systems are stored in the `TWA_DATA_DIR` directory, while binary files are stored in `TWA_home`.

Also, configuration settings, usually stored in the `server.xml` file, are now divided into several `.xml` files.

To modify Open Liberty configuration settings, first find out the `.xml` file to be modified and the directory where it is stored.

[Table 13: Open Liberty configuration files on page 200](#) lists the files available for Open Liberty configuration.

Table 13. Open Liberty configuration files

Configuration file	Functionality
<i>TWA_DATA_DIR</i> /usr/servers/engineServer/configDropins/overrides	
authentication_config.xml	Authentication settings
datasource.xml	Datasource settings
host_variables.xml	Hostname and port settings
jvm.options	Settings for Java Virtual machine, such as HeapSize
ports_variables.xml	Hostname and port settings
ssl_variables.xml	SSL connections and certificates
wauser_variables.xml	Authentication settings
<i>TWA_DATA_DIR</i> /usr/servers/engineServer/resources/security	
TWSServerKeyFile.p12	Open Liberty key store file, containing security keys
TWSServerTrustFile.p12	Open Liberty trust store file, containing certificates
ltpa.keys	LTPA keys, to be configured for Single Sign On

On Windows systems, there is no such separation and the path to Open Liberty configuration files is as follows:

On master domain managers

<TWA_home>\usr\servers\engineServer\configDropins\overrides

On Dynamic Workload Console

<DWC_home>\usr\servers\dwcServer\configDropins\overrides

Configuring the TLS V1.3 security protocol

The following procedures enable you to configure the TLS V1.3 security protocol for HCL Workload Automation. If you want to configure your environment with the TLS V1.3 protocol, it is recommended to use a 4k-length certificate.



Note: TLS V1.3 security protocol support is available from HCL Workload Automation version 10.1 FP4 onwards.

The configuration of the TLS V1.3 security protocol can be manually done on every component:

- [Dynamic agents on page 201](#)
- [Open Liberty on page 202](#)
- [Native components and fault-tolerant agents on page 202](#)

The configuration of the TLS V1.3 security protocol can only be set using custom certificates with RSA keys of at least 2K.

Dynamic agents

To enable the TLS V1.3 security protocol for dynamic agents you must open the `<TWSDATA>/ITA/cpa/ita/ita.ini` file and go to the *ITA SSL* section. Here you can set the security modifying the following keywords:

Enabling the TLS V1.3 security protocol exclusively

```
ssl version= TLSv1.3
ssl_ciphers=
```

Enabling the TLS V1.2 and TLS V1.3 security protocols

```
ssl version= atleast.TLSv1.2
ssl_ciphers=
```

where:

ssl_version

Specify the SSL version to be used. Supported values are:

- **atleast.TLSv1.0**
- **atleast.TLSv1.1**
- **atleast.TLSv1.2**
- **atleast.TLSv1.3**

where you specify the minimum version of the TLS protocol to be used. In this case, HCL Workload Automation uses the specified version of the protocol or a higher version, if supported.

- **max.TLSv1.0**
- **max.TLSv1.1**
- **max.TLSv1.2**
- **max.TLSv1.3**

where you specify the maximum version of the TLS protocol to be used. In this case, HCL Workload Automation uses the specified version of the protocol or a lower version.

- **TLSv1.0**
- **TLSv1.1**
- **TLSv1.2**
- **TLSv1.3**

where you specify the exact version of the TLS protocol to be used. In this case, HCL Workload Automation uses the specified version of the protocol.

ssl_ciphers

Define the ciphers that the workstation supports during an SSL connection.

If you want to use an OpenSSL cipher class, use the following command to find out the list of available classes:

```
openssl ciphers
```

For a full list of supported ciphers, see [SSL Ciphers](#) and [OpenSSL](#).



Note: The dynamic agents must be restarted after the modifications are completed.

Open Liberty

The following procedures must be repeated for every HCL Workload Automation component in the environment that has Open Liberty installed.

To enable the TLS V1.3 security protocol for Open Liberty you must copy the `<TWA_INSTALL_FOLDER>/usr/servers/engineServer/configDropins/defaults/ssl_config.xml` file and paste it in the following folders:

- `<TWA_INSTALL_FOLDER>/usr/servers/engineServer/configDropins/overrides`
- `<DWC_INSTALL_FOLDER>/usr/servers/dwcServer/configDropins/overrides`

You must then edit the `ssl_config.xml` file:

Enabling the TLS V1.3 security protocol exclusively

```
sslProtocol="TLSv1.3"
```

Enabling the TLS V1.2 and TLS V1.3 security protocols

```
sslProtocol="TLSv1.2,TLSv1.3"
```

No spaces can be used before or after the comma.



Note: Open Liberty must be restarted after the modifications are completed.

Native components and fault-tolerant agents

The following procedures must be repeated for every native component and fault-tolerant agents in the HCL Workload Automation environment.

To enable the TLS V1.3 security protocol for native components and fault-tolerant agents, you must open the `<TWSDATA>/localopts` file. Choose the procedure that applies to the kind of certificates you are using:

Opens SSL

Enabling the TLS V1.3 security protocol exclusively

Set the **ssl version** keyword as follows:

```
ssl version = TLSv1.3
```



Note: The native components and fault-tolerant agents must be restarted after the modifications are completed.

Using SSL for event-driven workload automation (EDWA) behind firewalls

This feature allows a domain manager to be run as a reverse proxy for HyperText Transfer Protocol (HTTP) and Event Integration Facility (EIF) protocols, forwarding traffic to the Event Processor. An option, enabled using the **optman** command-line program, allows you to choose if workstations that are behind a firewall must connect to the domain manager instead of to the event processor, causing the new proxy on the domain manager to forward its traffic to the event processor.



Restriction: This configuration is not supported if the agent workstation is a dynamic agent.

The incoming traffic is rerouted as follows:

- If an agent is behind a firewall, the traffic is routed to the domain manager on the agent. If an agent is not behind a firewall, the traffic is sent directly to the event processor.
- If domain managers have child nodes behind a firewall, the traffic is rerouted to the event processor.
- Primary domain managers always reroute traffic to the current event processor.
- Lower level domain managers reroute traffic to upper level domain managers if they are behind a firewall, or to the event processor if they are not behind a firewall.

To use this feature, perform the following steps:

1. Enable the feature by setting the **optman** option to `yes`. The default value is `no`:

```
enEventDrivenWorkloadAutomationProxy | pr = {yes|no}
```

2. In the workstation definition in the database for the agent, set the `behindfirewall` attribute to `ON`.
3. Configure OpenSSL on the domain manager.

For details about setting the `behindfirewall` attribute, see the section about workstation definition in *User's Guide and Reference*.

Configuring your master domain manager and dynamic domain manager in SSL mode

Configuring your master domain manager and dynamic domain manager in SSL mode

About this task

By default, starting from version 10.1 master domain manager and dynamic domain manager are installed in SSL mode.

If you are upgrading from a version earlier than 10.1 and want to set up your master domain manager and dynamic domain manager in SSL mode, perform the following steps:

1. Install the master domain manager or upgrade your current master domain manager to the latest version, for example version 10.2.
2. Stop Open Liberty, as described in the topic about starting and stopping the application server in *Administration Guide*.
3. Replace the values of the following parameters in the `localopts` file with the following values:
 - **nm SSL full port** = 31113
 - **SSL key** =TWA_home/TWS/ssl/OpenSSL/TWSClient.key
 - **SSL certificate** = TWA_home/TWS/ssl/OpenSSL/TWSClient.cer
 - **SSL key pwd** = TWA_home/TWS/ssl/OpenSSL/password.sth
 - **SSL CA certificate** = TWA_home/TWS/ssl/OpenSSL/TWSTrustCertificates.cer
 - **SSL random seed** =TWA_home/TWS/ssl/OpenSSL/TWS.rnd
 - **SSL Encryption Cipher** = TLSv1.2

For more information about the `localopts` file, see [Setting local options](#)

4. Modify the master domain manager and dynamic domain manager using the `composer mod` command, as follows:

```
CCPUNAME your_master_domain_manager_workstation
```

```
DESCRIPTION "MANAGER CPU"
```

```
OS UNIX
```

```
NODE localhost TCPADDR 31111
```

```
SECUREADDR 31113
```

```
DOMAIN MASTERDM
```

```
FOR MAESTRO
```

```
TYPE MANAGER
```

```
AUTOLINK ON
```

```
BEHINDFIREWALL OFF
```

```
SECURITYLEVEL FORCE_ENABLED
```

```
FULLSTATUS ON
```

```
END
```

```
CPUNAME your_broker_workstation
```

```
DESCRIPTION "This workstation was automatically created."
```

```
OS OTHER
```

```
NODE localhost TCPADDR 41114
```

```
SECUREADDR 41114
```

```
DOMAIN MASTERDM
```



```

FOR MAESTRO

    TYPE BROKER

    AUTOLINK ON

    BEHINDFIREWALL OFF

    SECURITYLEVEL FORCE_ENABLED

    FULLSTATUS OFF

END

```

5. Modify the **Broker.Workstation.PortSSL** parameter in the `BrokerWorkstation.properties` file from `false` to `true`.

The **Broker.Workstation.PortSSL** parameter specifies the port used by the broker server to listen to the incoming traffic (equivalent to the Netman port) in SSL mode. 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 HCL Workload Automation network (one with the master domain manager and one with every backup master domain manager you install) to ensure consistency when you switch masters.

6. Start Open Liberty, as described in the topic about starting and stopping the application server in *Administration Guide*.
7. Stop and start all HCL Workload Automation processes.
8. Run

```
Jnextplan -for 0000
```

Part III. Configuring

Configuring HCL Workload Automation components after installation.

About this task

You must configure HCL Workload Automation components after installation.

Setting the environment variables

About this task

Before you configure your HCL Workload Automation components, you must set the environment variables using the `twc_env` or `twc_env` script. You can use the two scripts interchangeably.

The `twc_env` script is located in the following paths:

On Windows operating systems

`HCL/TWA`

On UNIX operating systems

`/opt/HCL/TWA`

The upgrade installation process for agents installs a new version of the `twc_env` script in the directory `<TWA_HOME>/TWS`, where `<TWA_HOME>` is the HCL Workload Automation installation directory. A backup copy of your original version is created in a backup directory. After the upgrade process, merge the content of the new version with the content of the original version to carry your customized content into the new version.

The script is copied into the backup instance in `/<working_dir>/TWA_<user_name_of_installation_user>`

On Windows™ operating systems, run the `twc_env.cmd` shell script to set up both the `PATH` and `TWS_TISDIR` variables. For example, if HCL Workload Automation is installed in the `%ProgramFiles%\HCL\TWA\TWS` directory, the `<PATH>` variable is set as follows:

```
c:\Program Files\HCL\TWA\TWS;c:\Program Files\HCL\TWA\TWS\bin
```



Note: If you have more than one version of HCL Workload Automation installed on your computer, make sure `<TWS_TISDIR>` points to the latest one. This ensures that the most recent character set conversion tables are used.

On UNIX™ and Linux™ operating systems, source the `./twc_env.sh` shell script to set up the `PATH`, `TWS_TISDIR`, and `UNISONWORK` variables. For example, if HCL Workload Automation is installed in the default directory `/opt/HCL/TWA/TWS` directory, `./twc_env.sh` sets the variables as follows:

```
PATH=/opt/HCL/TWA/TWS:/opt/HCL/TWA/TWS/bin:$PATH
export PATH

TWS_TISDIR=/opt/HCL/TWA/TWS
export TWS_TISDIR
```

The `twc_env` script has two versions:

- `twc_env.sh` for Bourne and Korn shell environments
- `twc_env.csh` for C Shell environments

Chapter 8. Configuring a master domain manager

About this task

After you installed a master domain manager, follow the steps in this section to add the *FINAL* and *FINALPOSTREPORTS* job streams to the database.

The *FINAL* job stream is placed in production every day and runs JnextPlan before the start of a new day.

The *FINALPOSTREPORTS* job stream, responsible for printing post production reports, follows the *FINAL* job stream and starts only when the last job listed in the *FINAL* job stream (*SWITCHPLAN*) is completed successfully.

The installation creates the `<TWS_INST_DIR>\TWS\Sfinal` file that contains the *FINAL* and *FINALPOSTREPORTS* job stream definitions.

You can use the `<TWS_INST_DIR>\TWS\Sfinal` or create a customized new file for the *FINAL* job stream. For more information, see the section about customizing the final job stream in *User's Guide and Reference*.

The following steps give an example of how to configure a master domain manager after the installation:

1. Log in as *TWS_user* or as administrator.
2. Set the environment variables. See [Setting the environment variables on page 206](#).
3. Add the *FINAL* and *FINALPOSTREPORTS* job stream definitions to the database by running the following command from the `/opt/HCL/TWA/TWS` directory:

```
composer add Sfinal
```

where *Sfinal* is the name of the file that contains the *FINAL* and *FINALPOSTREPORTS* job stream definitions.

4. Add the *FINAL* and the *FINALPOSTREPORTS* job streams to the plan by running:

```
JnextPlan
```

You can automate this step after installation. See the section about automating production plan processing in *User's Guide and Reference*.

5. When JnextPlan completes, check the status of HCL Workload Automation:

```
conman status
```

If HCL Workload Automation started correctly, the status that is returned by the command is `Batchman LIVES`.

6. Change the workstation limit value to run jobs. The default job limit after installation is **0**, so no jobs run at any time. Raise the job limit to allow jobs to run, for example, to run 10 jobs at the same time:

```
conman "limit ;10"
```

If no workstation name is specified for the **limit** command, the default value is the current login workstation.



Note: If the priority of jobs is **HI** (100) or **GO** (101), the limit is ignored and the jobs run even if the limit is 0, unless the workstation fence is greater than or equal to the priority.

Additionally, the following configuration procedures might be necessary. For information about these procedures, see the relevant sections in *Administration Guide*:

- Customizing and configuring global, local, and user options.
- Customizing and configuring user authentication to allow users authorization on actions and objects, and to configure LDAP.
- Setting connection security to enable SSL for inter-component communications.

Chapter 9. Configuring a master domain manager configured as backup

About this task

After you install a master domain manager configured as backup, perform the following additional configuration steps:

1. Log in as `TWS_user` on your master domain manager.
2. Add the username and password for the master domain manager configured as backup to the `useropts` file. For details, see the *Administration Guide* section about setting user options..
3. Set the environment variables by running `twc_env` as described in [Setting the environment variables on page 206](#).
4. Define the master domain manager configured as backup as a full status autolink fault-tolerant agent in the HCL Workload Automation database, using the `composer` command interface or the Dynamic Workload Console. In this example with `composer`, type the following command:

```
composer  
new
```

5. Type the workstation definition in the text editor, for example:

```
CPUNAME BDM1  
DESCRIPTION "Backup master domain manager"  
OS UNIX  
NODE lab777  
TCPADDR 31111  
FOR MAESTRO  
  TYPE FTA  
  AUTOLINK ON  
  BEHINDFIREWALL OFF  
  FULLSTATUS ON  
end
```

For more information about workstation definitions, see the section about workstation definition in *User's Guide and Reference*.

6. Run `JnextPlan -for 0000` to include the master domain manager configured as backup workstation in the plan and to send the Symphony™ file to it.



Note: Ensure that the global option `carryforward` is set to `all`, otherwise only incomplete job streams are carried forward.

7. Change the workstation limit to allow jobs to run on the workstation. For example, set the number of jobs to run concurrently on the workstation to `10`:

```
conman "limit DM1;10"
```



Note: If you are logged into the master domain manager configured as backup, the workstation name (`DM1` in the above example) is not required.

Additionally, the following configuration procedures might be necessary. For information about these procedures, see the relevant sections in *Administration Guide*:

- Customizing and configuring global, local, and user options.
- Customizing and configuring user authentication to allow users authorization on actions and objects, and to configure LDAP.
- Setting connection security to enable SSL for inter-component communications.

Chapter 10. Configuring a domain manager

About this task

After you install a domain manager, perform the following configuration steps:

1. Log in as *TWS_user* on your master domain manager.
2. Set the environment variables by running *twc_env* as described in [Setting the environment variables on page 206](#).
3. Create a new domain by running the following command:

```
composer new domain
```

4. Type the domain definition in the text editor, for example:

```
DOMAIN DOMAIN1
  DESCRIPTION "Sample Domain"
  PARENT MASTERDM
END
```

5. Define the domain manager as a full status autolink fault-tolerant agent in the HCL Workload Automation database, using the composer command interface or the Dynamic Workload Console. In this example, using composer, type:

```
composer
new
```

6. Type the workstation definition in the text editor, for example:

```
CPUNAME DDM1
  DESCRIPTION "domain manager"
  OS UNIX
  NODE lab0777
  TCPADDR 31111
  DOMAIN DOMAIN1
  FOR MAESTRO
    TYPE MANAGER
    AUTOLINK ON
    BEHINDFIREWALL OFF
    FULLSTATUS ON
END
```

For more information about workstation definitions, see the section about workstation definition in *User's Guide and Reference*.

7. Run **JnextPlan -for 0000** to include the domain manager workstation in the plan and to send the Symphony file to it.



Note: Ensure that the global option carryforward is set to all, otherwise only incomplete job streams are carried forward.

8. Change the workstation limit to allow jobs to run on the workstation. For example, set the number of jobs to run concurrently on the workstation to 10:

```
conman "limit;10"
```


Chapter 11. Configuring a backup domain manager

About this task

After you install a backup domain manager, perform the following configuration steps:

1. Log in as *TWS_user* on your master domain manager.
2. Set the environment variables by running *twc_env* as described in [Setting the environment variables on page 206](#).
3. Define the backup domain manager as a full status autolink fault-tolerant agent in the HCL Workload Automation database, using the *composer* command interface or the Dynamic Workload Console. In this example, using *composer*, type:

```
composer new
```

4. Type the workstation definition in the text editor, for example:

```
CPUNAME Backup_DM
DESCRIPTION "backup domain manager"
OS UNIX
NODE lab0777
TCPADDR 31111
DOMAIN MDM
FOR MAESTRO
  TYPE FTA
  AUTOLINK ON
  BEHINDFIREWALL OFF
  FULLSTATUS ON
END
```

For more information about workstation definitions, see the section about workstation definition in *User's Guide and Reference*.

5. Run **JnextPlan -for 0000** to include the backup domain manager workstation in the plan and to send the Symphony file to it.



Note: Ensure that the global option *carryforward* is set to *all*, otherwise only incomplete job streams are carried forward.

6. Change the workstation limit to allow jobs to run on the workstation. For example, set the number of jobs to run concurrently on the workstation to 10:

```
conman "limit;10"
```

Chapter 12. Configuring a dynamic domain manager

About this task

After you install a dynamic domain manager, perform the following configuration steps:

1. Log in as *TWS_user* on your master domain manager.
2. Set the environment variables by running `twc_env` as described in [Setting the environment variables on page 206](#).
3. Run **JnextPlan -for 0000** to include the dynamic domain manager workstation in the plan and to send the Symphony file to it.



Note: Ensure that the global option `carryforward` is set to `all`, otherwise only incomplete job streams are carried forward.

4. Change the workstation limit to allow jobs to run on the workstation. For example, set the number of jobs to run concurrently on the workstation to 10:

```
conman "limit;10"
```

Chapter 13. Configuration steps for a dynamic domain manager configured as backup

About this task

After you install a dynamic domain manager as backup, perform the following configuration steps:

1. Log in as *TWS_user* on your master domain manager
2. Set the environment variables by running *twc_env* as described in [Setting the environment variables on page 206](#).
3. Define the dynamic domain manager as backup as a full status autolink fault-tolerant agent in the HCL Workload Automation database, using the composer command interface or the Dynamic Workload Console. In this example using composer, type:

```
composer  
new
```

4. Type the workstation definition in the text editor, for example:

```
CPUNAME BDDM1  
DESCRIPTION "backup dynamic domain manager"  
OS UNIX  
NODE lab00777  
TCPADDR 31111  
DOMAIN DYNAMICDM  
FOR MAESTRO  
  TYPE FTA  
  AUTOLINK ON  
  BEHINDFIREWALL OFF  
  FULLSTATUS ON  
END
```

For more information about workstation definitions, see the section about workstation definition in *User's Guide and Reference*.

5. Run **JnextPlan -for 0000** to include the dynamic domain manager as backup workstation in the plan and to send the Symphony file to it.



Note: Ensure that the global option *carryforward* is set to *all*, otherwise only incomplete job streams are carried forward.

6. Change the workstation limit to allow jobs to run on the workstation. For example, set the number of jobs to run concurrently on the workstation to 10:

```
conman "limit;10"
```

Chapter 14. Configuring a dynamic agent

How to configure a dynamic agent.

About this task

The dynamic agent installation process automatically adds the workstation definition to the database and registers the workstation definition to the dynamic workload broker installed on the master domain manager or dynamic domain manager that you chose during the installation process.

Dynamic agents can be organized in pools to help organize your environment based on the availability of workstations and on the requirements of the jobs that need to be run. You can create a pool, adding dynamic agents to a workstation definition of type pool, or, you can automatically register agents to pools through a different process. See the topic about automatically registering agents to a pool in the *Planning and Installation Guide*.

After installing a dynamic agent, depending on the `enAddWorkstation` global option settings in the master domain manager, perform the following steps:

If `enAddWorkstation` is set to **no**:

1. Run JnextPlan with the **-for 0000** option to add the dynamic agent workstation definition to the plan and to send the Symphony file to it. For more information about workstation definitions, see the section about workstation definition in *User's Guide and Reference*.



Note: To carry forward completed and incomplete job stream instances, ensure that the `carryforward` global option is set to `all` or run JnextPlan **-for 0000** with the **-noremove** option.

2. Change the workstation limit to allow jobs to run on the workstation. For example, set the number of jobs that can run concurrently on the workstation to 10:

```
conman "limit DA235007_00;10"
```

If `enAddWorkstation` is set to **yes**:

The workstation definition is automatically added to the plan after it is defined in the database by the installation process. The `workstationLimit` global option specifies the dynamic agent workstation limit value that the dynamic agent workstation assumes after the workstation is added to the plan.

For more information about how to modify the `enAddWorkstation` and `workstationLimit` global option settings, see the section about global options settings in *Administration Guide*.

For more information about troubleshooting, see the section about troubleshooting when automatically adding dynamic agent workstations to the plan in *Troubleshooting Guide*.

You might also need to run the following configuration procedures. For information about these procedures, see *Administration Guide*.

- Customizing and configuring `jobmanager.ini` and user options.
- Customizing and configuring `JobManagerGW.ini` for opening communication between the gateway and the dynamic workload broker.
- Customizing and configuring user authentication to allow user authorization for actions and objects, and to configure LDAP.
- Setting connection security to enable SSL for inter-component communications.

Automatically register agents to pools

The dynamic agent installation process automatically adds the workstation definition to the database and registers the workstation definition to the dynamic workload broker installed on the master domain manager or the dynamic domain manager that you specify during the installation process.

You can add dynamic agents in pools to help organize your environment based on the availability of workstations and the requirements of the jobs to be run. Normally, when you create a pool, you add the dynamic agents to a workstation definition of type pool.

Starting from HCL Workload Automation version 9.4 Fix Pack 4, you can automatically register dynamic agents in pools by editing the `pools.properties` file located in `TWS_home>/ITA/cpa/config`.

Starting from version 9.5, the `pools.properties` file is located in the following paths:

On Windows operating systems

```
<TWS_home>\ITA\cpa\config
```

On UNIX operating systems

```
<TWA_DATA_DIR>/ITA/cpa/config
```

This alternative way of registering dynamic agents to a pool can be useful when you need to quickly add more than one agent to a pool, or when you want to associate multiple pools to a dynamic agent.

The file is composed by a series of lines with a list of pools to which the agent will be automatically registered. To make the changes in this file effective on the agent, you must stop the agent, edit the file, then start the agent. See the section about the `ShutDownLwa` and `StartUpLwa` commands in *User's Guide and Reference*.

For example, if you want to register a dynamic agent with three different pools, then edit the `pools.properties` file as follows:

```
P00L1
P00L2
P00L3
```

By default, master domain manager and backup domain manager dynamic agents register with the pool named `MASTERAGENTS`. In this case, the `pools.properties` file on these agents contains the following default entry:

```
$MASTERAGENTS
```



Note: The default name for this pool workstation, MASTERAGENTS, can be modified using the optman global option `resubmitJobName`. See the detailed description of the global options in the *Administration Guide* for details about this option.

The following options are supported for each entry in the `pool.properties` file:

;skip

Use this option to exclude pools from even being considered. You might want to ignore specific pools for a period of time, but still maintain them in the list so that they can be considered in the future.

;optional

Use this option to specify that a pool is not obligatory, but optional, so that if the agent is unable to register to a pool, for example, a pool no longer exists) then the pool is ignored.

If an agent has obligatory pools in the `pools.properties` file that are not defined in the system, then the agent will not be able to automatically register and go online. To ensure agent connectivity, these options can be used to manage situations where the agent needs to online even if some pools are not defined.

If the agent does not receive any errors, then the agent goes online and is added to all of the pools in the list, except for those with the `;skip` option specified.

If, instead, the agent encounters an error, the agent is able to determine which of the pools in the list has a problem. If the problematic pool is mandatory (without the `;optional` option specified), then the agent goes offline and is not added to any of the pools. If the problematic pool is optional (with the `;optional` option specified), the pool is discarded.

To demonstrate how you can use these options in the `pool.properties` file, consider the following example:

```
$MASTERAGENTS;optional
POOL1
POOL2;skip
POOL3;optional;skip
POOL4;optional
```

Case 1: POOL1 and POOL4 exist, MASTERAGENTS does not exist

- POOL2;skip is not considered at all.
- POOL3;optional;skip is not considered at all because the `;skip` option overrides the `;optional` option.
- MASTERAGENTS;optional is the problematic pool and is optional and therefore not considered by the agent.
- POOL1 is not a problematic pool.
- POOL4 is not a problematic pool.

Outcome: The agent goes online and is inserted in POOL1 and POOL4.

Case 2: POOL1 does not exist, MASTERAGENTS and POOL4 exist

- POOL2;skip is not considered at all.
- POOL3;optional;skip is not considered at all because the ;skip option overrides the ;optional option.
- MASTERAGENTS;optional is not a problematic pool.
- POOL1 is the problematic pool and is mandatory and cannot be discarded.
- POOL4 is not a problematic pool.

Outcome: The agent goes offline and is not inserted in any of the pools.

Revoking and reissuing a JSON Web Token

Steps to revoke and reissue a JWT

To revoke a JSON Web Token (JWT), delete the workstation definition to which the JWT is associated from the database. You can perform this operation from the Dynamic Workload Console or from the command line. To delete the agent from the command line, perform the following steps:

1. Open a shell session.
2. Launch the composer script.
3. Type the following command:

```
delete workstation workstation_name
```

where

workstation_name

is the name of the agent whose JWT you want to revoke.

For more information about the delete command, see the section about the delete command in *User's Guide and Reference*.

From the Dynamic Workload Console, you can perform the same operation as follows:

1. Log in to the Dynamic Workload Console.
2. Click **Design > Workload Designer**.
3. Select an engine.
4. Click the **Workstation** item card to display all existing workstations.
5. Select the workstation to be deleted.
6. Click **Delete**.

If you want the agent to authenticate with JWT again, download a new JWT to the agent using the `AgentCertificateDownloader` script.

Consider the following example:

```
./AgentCertificateDownloader.sh --jwt true --work_dir /tmp --tdwbhostname <broker_host_name>
--tdwbport <broker_port> --gwid <gateway_id> --gateway local -apikey <apikey>
```

For more information about the AgentCertificateDownloader script, see [Certificates download to dynamic agents and fault-tolerant agents - AgentCertificateDownloader script on page 482](#).

Chapter 15. Configuring a remote command-line client

About this task

To configure a remote command-line client, 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 the remote command-line client is installed with a fault-tolerant agent.
2. Open the `localopts` configuration file in the fault-tolerant agent instance.
3. Complete the `# Attributes for CLI connections` configuration section to connect the remote command-line client to the command-line server in the master domain manager:

HOST

The IP address or host name of the workstation where the master domain manager is installed.

PROTOCOL

The protocol that is used by the command-line client to connect to the workstation where the master domain manager is installed. The possible values are `http` and `https`. The default protocol that is used by the command-line client to establish a connection with the master is `https`.

PORT

The HTTP or HTTPS port number that is used to connect to the workstation where the master domain manager is installed. This port number must match the values that are defined for the master domain manager instance.

TIMEOUT

The timeout in seconds to wait for a master domain manager response.

CLISLSSERVERAUTH

Specify whether or not the connection to the master domain manager is SSL or not. If you set this value to `true`, perform the steps described in [Configuring SSL connection between remote command-line client and master domain manager on page 222](#).

CLISLSSERVERCERTIFICATE

Specify only if `CLISLSSERVERAUTH` is set to `true`. The absolute path of the `.arm` file of the server public certificate. For more information about this value, see [Configuring SSL connection between remote command-line client and master domain manager on page 222](#).

CLISLSTRUSTEDDIR

Specify only if `CLISLSSERVERAUTH` is set to `true`. The path of all the `.arm` files that the remote CLI must trust. For more information about this value, see [Configuring SSL connection between remote command-line client and master domain manager on page 222](#).

DEFAULTWS

The master domain manager workstation name.

USEROPTS

The file that contains the user name and password to use to connect to the master domain manager workstation. This user must be a valid user that is listed in the `Security` file on the master domain manager.

4. Save the `localopts`.
5. Restart the fault-tolerant agent processes to accept the `localopts` changes.

Configuring SSL connection between remote command-line client and master domain manager

Before you begin

Before starting with the procedure to configure the SSL connection between the remote command-line client and the master domain manager, ensure that you set the `CLISLSSERVERAUTH` property to `true` in the `localopts` file of the fault-tolerant agent instance.

About this task

To configure a remote command-line client to connect to a master domain manager in SSL mode, perform the following steps:

1. Extract the certificate on the master domain manager instance by running the following procedure:
 - a. Log on as Administrator on Windows operating systems, or as root on UNIX and Linux operating systems, on the machine where the master domain manager is installed.
 - b. Extract the `server.crt` base 64 certificate by running:

```
keytool -export
-alias server
-rfc
-file server.crt
-keystore <path>/TWSServerKeyFile.p12
-storepass default
```

where `<path>` is one of the following:

On Windows systems

```
<TWA_home>\usr\servers\engineServer\resources\security
TWSServerKeyFile.p12
```

On UNIX systems

```
<TWA_DATA_DIR>/usr/servers/engineServer/resources/security/
TWSServerKeyFile.p12
```

2. Log on as Administrator on Windows operating systems, or as root on UNIX and Linux operating systems, on the machine where the remote command-line client is installed with a fault-tolerant agent.

3. Perform a binary FTP of the `server.crt` certificate from the machine where you installed the master domain manager instance to the machine where you installed the remote command-line client in the directory `<FTA_INST_DIR>\ssl`.
4. Rename the `<FTA_INST_DIR>\ssl\server.crt` file to `<FTA_INST_DIR>\ssl\server.arm`.
5. Open the `localopts` configuration file in the fault-tolerant agent instance.
6. Complete one of the following attributes in the `# Attributes for CLI connections` configuration section and perform the actions:

CLISLSEVERCERTIFICATE

Specify the absolute path of the `server.arm` file on the fault-tolerant agent machine. In this example, `<FTA_INST_DIR>\ssl\server.arm`.

CLISLTRUSTEDDIR

Specify the path of the directory that contains all the `certificates.arm` files also the `<FTA_INST_DIR>\ssl\server.arm` that the remote command-line client can trust.



Note: Do not set simultaneously the `CLISLSEVERAUTH` and `CLISLTRUSTEDDIR` values. For more information about the SSL configuration, see the section about configuring secure communications in *Administration Guide*.

7. Save the `localopts` file.
8. Restart the fault-tolerant agent processes to accept the `localopts` changes.

Chapter 16. Configuring a z-centric agent on Windows operating systems

About this task

After you install a z-centric agent on a Windows operating system with a local or domain account, perform the following configuration steps:

1. Stop the dynamic agent.
2. From the **Start** menu, click **Administrative Tools > Services**.
3. Edit the properties of the following service by double-clicking on its name: `HCL Common Platform Agent: tws_cpa_agent_TWS_user`, where `TWS_user` is the name of the user for which HCL Workload Automation was installed (the name you supplied during installation).
4. Click label **Log On**.
5. Click **Log on as: Local System account**.
6. If you plan to run interactive jobs, check mark **Allow service to interact with desktop**.
7. Click **OK**.
8. From the **Start** menu, click **Administrative Tools > Local Security Policy**.
9. Remove the following permissions from the user created when you installed the z-centric agent:
 - Act as part of the operating system.
 - Log on locally.
 - Log on as batch.
10. Restart the dynamic agent.

Chapter 17. Adding a feature

About this task

Use the **twinsinst** script to add the following feature to the HCL Workload Automation agent in your distributed or end-to-end network:

Add the Java™ run time to an agent

During the installation or the upgrade of the agent you might have chosen not to add the Java™ run time that supports the running of job types advanced options. This option provides your agent with the following capabilities:

- Run job types with advanced options, both those types supplied with the product and the additional types implemented through the custom plug-ins.
- Enable the capability to run remotely, from the agent, the dynamic workload broker resource command on the server.

If you later decide that you require this function, you can add the Java™ run time separately, as described below.

Procedure

About this task

To modify agents by using the **twinsinst** script, perform the following steps:

On Windows™ operating systems

1. Download the elmage for your operating system. See [Downloading installation images on your workstation on page 232](#).
2. Log in as administrator on the workstation where you want to upgrade the product.
3. From the *root/TWS/operating_system* directory of the elmage, run **twinsinst** by using the synopsis described below.



Note: **twinsinst** for Windows™ is a Visual Basic Script (VBS) that you can run in CScript and WScript mode, for example:

```
cscript twinsinst -modify -uname username  
-password user_password -acceptlicense yes  
-addjruntime true
```

On UNIX™ and Linux™ operating systems

1. Download the elmage according to the operating system. See [Downloading installation images on your workstation on page 232](#).
2. From the *root/TWS/operating_system* directory, run the **twinsinst** script by using the synopsis described below.

A successful modify by using the **twinst** script issues the return code RC = 0. If the operation fails, to understand the cause of the error, see [Analyzing return codes for agent installation, upgrade, restore, and uninstallation on page 400](#).

Synopsis:

On Windows™ operating systems:

-acceptlicense *yes/no*

Specify whether or not to accept the License Agreement.

-addjruntime *true*

Adds the Java™ run time to run job types with advanced options to the agent. The run time environment is used to run application job plug-ins on the agent and to enable the capability to run remotely, from the agent, the dynamic workload broker resource command on the server. With the `-modify` option, the only valid value for this parameter is **true**.

This option is applicable to both fault-tolerant agents and dynamic agents.

-inst_dir *install_directory*

The installation directory for HCL Workload Automation. The default is the home directory of the user for which HCL Workload Automation is being installed.

-modify

Modifies an existing agent that was installed by using **twinst**.

-password *user_password*

Windows™ operating systems only. The password of the user for which you are upgrading HCL Workload Automation.

-recovInstReg *boolean*

Select this option to recover workstations that have corrupt registry files without reinstalling the product. If you specify this option, HCL Workload Automation re-creates the installation registries. Valid values are **true** and **false**. The default value is **false**.

You can use this option also to recover registry files in a cluster environment; in this case you can run the command on any node of the cluster and not necessarily on the node where you installed HCL Workload Automation. This is useful when the cluster node where the product is installed is unavailable or in an inconsistent state.

-uname *username*

The name of the user for which HCL Workload Automation is being updated. The software is updated in this user's home directory. This user name is not to be confused with the user that performs the upgrade.

Part IV. Upgrading

How to upgrade HCL Workload Automation to the current version.

Overview

When upgrading your HCL Workload Automation environment, it is a good practice to start with the upgrade of the Dynamic Workload Console first. If you upgrade the console to the new product version level, you can then use it to verify that your environment is working after upgrading the remaining components.

The upgrade procedure varies depending on the product version you currently have installed:

- if you have installed version 9.5.0.x or 10.x.x and want to upgrade to the General Availability version with a **direct upgrade** procedure, see [Performing a direct upgrade from v 9.5.0.x or v 10.x.x to v 10.2.5 on page 238](#).
- if you have installed version 9.5.0.x or 10.x.x and want to upgrade to the General Availability version with a **parallel upgrade** procedure, see [Parallel upgrade from version 9.5.0.x or 10.x.x to version 10.2.5 on page 257](#). This procedure might be useful when you have some of your components installed on operating systems which are no longer supported in version 10.2.5 and therefore cannot perform a direct upgrade.
- if you have installed version 9.4.0.x and want to upgrade to version 10.2.5. In this case, only a **parallel upgrade** is supported. For more information, see [Parallel upgrade from version 9.4.0.x to version 10.2.5 on page 311](#).

In a **direct upgrade procedure from version 9.5.0.x or 10.x.x**, you upgrade the Dynamic Workload Console and its database, then upgrade the dynamic domain manager and its backups and the database, then master domain manager and its backups and the database, and finally the domain managers and their backups, and the agents.

In a **parallel upgrade procedure from version 9.5.0.x or 10.x.x**, you upgrade WebSphere Application Server Liberty, upgrade the Dynamic Workload Console and its database, then upgrade the database for the server components and install a new dynamic domain manager and master domain manager configured as a backup, then switch them to become the master. You then upgrade agents and domain managers.

In a **parallel upgrade procedure from version 9.4.0.x**, you install the Dynamic Workload Console at v 10.2.5. You then upgrade the database tables for the server components and their backups and install a new backup dynamic domain manager, switch the manager to the new backup, install a new backup and switch back the manager capabilities, so that the newly installed backup dynamic domain manager becomes the current dynamic domain manager.

You then proceed to running the serverinst script to install a version 10.2.5 master domain manager configured as a backup. The installation process is able to detect the presence of an existing master domain manager and automatically configures the second one as the backup master domain manager. The new backup master domain manager is configured to point to the existing database instance. You then perform a switch with the previous version master domain manager, so that the newly installed backup master domain manager becomes the current active master domain manager.

You then install a second master domain manager to act as the new backup master domain manager. Each Dynamic Workload Console, backup dynamic domain manager, dynamic domain manager, master domain manager and backup master domain manager installation requires its own installation of Open Liberty. The upgrade process concludes with upgrading agents. Agents can be upgraded with minimal disruption to scheduling activities.

Using the new features introduced with the latest release creates new records in the database which are not compatible with previous versions and therefore you cannot roll back your environment to a previous version.

If you upgrade HCL Workload Automation to version 10.2.x, and the HCL Workload Automation database was created with DB2, change the DB2 configuration parameter EXTENDED_ROW_SZ to ENABLE, or create a new buffer pool and table space with a page size of 16 kilobytes and migrate the tables to the new table space. For more information, see [Error in upgrading the HCL Workload Automation database when using a DB2 database on page 406](#).

Before upgrading, ensure that you have stopped workload processing on the master domain manager.

If you have previously customized the `tws_env` script, merge your changes into the new version of the script. Ensure you do not overwrite the parameters related to OpenSSL libraries during the merge.

Starting from version 10.2.5, FIPS mode is supported. HCL Workload Automation checks your FIPS settings in the source environment and applies the same settings when performing the upgrade. If FIPS was enabled in your source environment, it is automatically enabled in the target environment during the upgrade phase. However, if certificates are not secure by current FIPS 140-3 standards, the upgrade stops. For more information about managing FIPS settings, see [Enabling and disabling FIPS on page 379](#).

Choosing how to upgrade your network

After upgrading the Dynamic Workload Console, there are different approaches to upgrading the remaining components in your HCL Workload Automation environment. Because HCL Workload Automation supports compatibility with earlier versions, after upgrading the console, you can decide to proceed with upgrading in one of the following ways, depending on the type of your network:

Top-down

Upgrade components in the following order:

1. backup domain managers and domain managers
2. dynamic domain managers
3. backup master domain manager
4. master domain manager
5. agents

This order ensures that events involving folders are correctly managed by the master domain manager and sent to agents at a supported version level.

When you have a backup master domain manager at the V9.5 Fix Pack 2, or later, but the master domain manager is still at a previous product version level, problems can occur when monitoring objects that support the definition in a folder such as, prompts, workstations, and resources, as well as objects that contain the workstation in their object identifier, for example, job streams. More specifically these objects are not displayed in the results of the monitoring query on the plan if you use filters in your query. To solve this problem, upgrade the master domain manager to the V9.5 Fix Pack 2 level, or later, and then run `planman resynch`.

Many of the new functions that are introduced in the current version become available for each agent as it is upgraded. The disadvantage is that the same functions are not available to all agents at the same time.

Bottom-up

Upgrade components in the following order:

1. agents
2. backup domain managers and domain managers
3. dynamic domain managers
4. backup master domain manager
5. master domain manager

The new functions that are introduced in the current version are not available until the whole network is upgraded.

In the typical upgrade procedures documented in this manual, the top-down order is used.



Note: 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 this version, update the `Security` file before starting the HCL Workload Automation 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
.....
```

For details, see the section about configuring security file in *Administration Guide*.

Migrating custom events

When you perform an upgrade, custom events are not migrated. Therefore you must add custom events by following the manual procedure described below:

1. On the master domain manager, create a new XML file `<file_name>` where you can save custom events:

```
$ evtdef dumpdef <file_name>
```

2. Run the `switchmgr` command to switch from the master domain manager to the backup master domain manager.
3. Copy the XML file created in step 1 on the backup master domain manager.
4. Load the custom event definition on the backup master domain manager by running the following command:

```
$ evtdef loaddef <file_name>
```

Where `<file_name>` is the name of the XML file that you copied from the master domain manager and saved on the backup master domain manager.

5. Stop WebSphere Application Server Liberty on the Dynamic Workload Console.

6. Start WebSphere Application Server Liberty on the Dynamic Workload Console by running the following command:

```
<DWC_HOME>/appservertools/startAppServer.sh -directclean
```

Chapter 18. Downloading installation images on your workstation

Steps to take when downloading images on your workstation.

About this task

Complete the following procedure to download the installation images to upgrade your environment to the latest level:

1. Ensure that your workstation has sufficient space to store both the files you download from [HCL Software](#) and the extracted installation image. For more information about system requirements, see and [Dynamic Workload Console Detailed System Requirements](#). To install the product, download all the required images from [HCL Software](#). The zip contains both the General Availability 10.2.5 image and the latest fix pack image, if available.
2. Download the installation images from [HCL Software](#).
3. Extract the installation image from the downloaded file and verify that the installation image is complete.

Chapter 19. Upgrading from the CLI

Upgrade HCL Workload Automation from the command-line interface.

The upgrade procedure varies depending on the product version you currently have installed:

- if you have installed version 9.5.0.x or 10.x.x and want to upgrade to the General Availability version with a **direct upgrade** procedure, see [Performing a direct upgrade from v 9.5.0.x or v 10.x.x to v 10.2.5 on page 238](#).
- if you have installed version 9.5.0.x or 10.x.x and want to upgrade to the General Availability version with a **parallel upgrade** procedure, see [Parallel upgrade from version 9.5.0.x or 10.x.x to version 10.2.5 on page 257](#). This procedure might be useful when you have some of your components installed on operating systems which are no longer supported in version 10.2.5 and therefore cannot perform a direct upgrade.
- if you have installed version 9.4.0.x and want to upgrade to version 10.2.5. In this case, only a **parallel upgrade** is supported. For more information, see [Parallel upgrade from version 9.4.0.x to version 10.2.5 on page 311](#).

Certificates are now required when installing or upgrading HCL Workload Automation. You can no longer install nor upgrade HCL Workload Automation without securing your environment with certificates. The required certificates are:

- ca.crt
- tls.key
- tls.crt

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

Before upgrading

Before starting to upgrade the product, verify that your network has the minimum required supported versions of the operating system, product, and database.

Supported operating systems

To obtain an updated list of the supported operating systems, see the **Supported Operating Systems Report**, available in Product Requirements.

For a complete list of system requirements (disk spaces, temporary spaces and RAM usage), see [HCL Workload Automation Detailed System Requirements](#).

Supported databases

For an up-to-date list of supported databases, see the **Supported Software Report**, available in Product Requirements.

Product level prerequisites for master domain manager and its backup, dynamic domain manager and its backup, and agents

Before you start the upgrade, verify that your environment has the required product level prerequisites. For a complete list of product level prerequisites, see [HCL Workload Automation Detailed System Requirements](#).

User authorization requirements

Before starting to upgrade, verify that the user running the process has the following authorization requirements:

Windows™ operating systems

If you set the Windows User Account Control (UAC), your login account must be a member of the Windows™ **Administrators** group or domain administrators with the right, **Act as Part of the Operating System**.

If you set the Windows User Account Control (UAC) on the workstation, you must run the installation as **administrator**.

UNIX™ and Linux™ operating systems

If the component was installed with root privileges, **root** access is required. If you performed a **no-root installation**, specify the same user used for installing the component.

SSL mode configuration

If the HCL Workload Automation environment is configured in SSL mode, ensure one of the following conditions is met in the `localopts` file before you upgrade master domain manager, backup master domain manager, dynamic domain manager, or fault-tolerant agents to Version 10.2.5 or later:

- the **ssl version** parameter is set to `atleast.TLSv1.2` OR
- the **ssl cipher** parameters is set to a high value.

For more information about the `localopts` file, see the section about setting local options in *Administration Guide*.

Securing your environment with certificates

Starting from version 10.2.1, using certificates is mandatory when installing or upgrading the product. You can use the `certman` command to extract certificates from an existing keystore, generate new certificates from a new or existing CA, and much more. For more information, see the topic about managing certificates using `certman` in *Administration Guide*.

Upgrading to 10.1 Fix Pack 1 or later using custom certificates

In 10.1 FP1 version, the JWT feature has been introduced. Performing an upgrade of the master domain manager to 10.1 FP1 from any previous version, can potentially cause problems with JWT functionality if the master domain manager is using custom certificates with a custom label.

The new elements listed above identify the certificate using the **server** label, instead of the custom label defined in the **keyName** properties. This prevents WebSphere Application Server Liberty Base from signing new JWTs.

To work around this problem, perform the following steps:

1. Update the **<jwtBuilder>** elements by modifying the **keyAlias** property to the correct value.
2. To verify the signature of a JWT received in a connection from another entity, WebSphere Application Server Liberty Base retrieves the public information associated to the certificate from the **<WA_DATA>/usr/servers/engineServer/resources/security/TWSServerTrustFile.p12** file. You can find the public information in the **keyName="{mp.jwt.trust.key}"** property within the **<mpJwt>** elements. These elements use a variable which is declared within the new **jwt_variables.xml** file that is created in the **overrides** folder after the upgrade:

```
<server description="jwt_variables">

    <variable name="mp.jwt.trust.key" value="twstrustkey"/>

</server>
```

3. Also, add the public information only of the custom certificate in the **TWSServerTrustFile.p12** file, under that alias (overwriting the already existing one).
4. Alternatively, it is possible to add it as a new entry with a new label, but the **jwt_variables.xml** file should be updated accordingly. For more information, see [Enabling API Key authentication after upgrading on page 369](#).
5. The agent must have the public information associated to the certificate used by the master domain manager when creating a new JWT. The reason for this is that also the agent needs to verify the signature of a JWT received from the master domain manager. Therefore, it is required to also add the public information only of the custom certificate of the master domain manager (the file that was added in the **TWSServerTrustFile.p12** file on the master domain manager) in the **TWSClientKeyStore** file of the agent.

Support for OpenSSL 3.0.x libraries from UNIX operating systems

If you install the master domain manager on recent UNIX operating systems, you can use the OpenSSL 3.0.x libraries provided with the operating system. The list of UNIX operating systems whose libraries you can use is as follows:

- Ubuntu 22
- AIX 7.3
- Red Hat 9

To ensure HCL Workload Automation uses these libraries, always launch the installation or upgrade procedure from a brand new shell. You can also check the OpenSSL library currently in use with the **which openssl** command and check the OpenSSL version with the **openssl version** command.

Downloading installation images

Before starting to upgrade, download the installation images. For further information, see [Downloading installation images on your workstation on page 232](#)

Scanning system prerequisites for HCL Workload Automation

Before installing or upgrading the product, HCL Workload Automation automatically runs a scan on your system.

Before you begin

When installing HCL Workload Automation using the `serverinst` script, the script first runs the scanner to verify system prerequisites. For more information about prerequisites, see the topic about Product Requirements in the online documentation.



Note: To ensure that the prerequisite scan process does not fail, verify that the `bc` executable is present on the local system and that it is set in the PATH environment variable. If you do not want to install the `bc` executable, you can skip the prerequisites check by using the `skipcheckprereq` parameter when running the `serverinst` and `twinst` parameters. For more information about the `bc` executable, see [bc, an arbitrary precision calculator language](#). For more information about installation commands, see [Server components installation - serverinst script on page 442](#) and [Agent installation parameters - twinst script on page 119](#).

About this task

Having an environment that meets the product system requirements ensures that an installation or upgrade succeeds without any delays or complications.

The scan verifies that:

- The operating system is supported for the product.
- On UNIX™ operating systems, the necessary product libraries are installed.
- There is enough permanent and temporary disk space to install both the product and its prerequisites.
- There is enough memory and virtual memory.



Note: The scan verifies only that the environment meets the requirements of HCL Workload Automation. It does not check the requirements for other components, such as DB2®.

If any of these checks fails, HCL Workload Automation returns an error message.

The log files for the server components are located in:

On Windows™ operating systems:

```
<TWA_home>\logs\serverinst<version_number>.log
```

On UNIX™ and Linux™ operating systems:

```
<TWA_DATA_DIR>/installation/logs/serverinst<version_number>.log
```

The log files for the Dynamic Workload Console are located in:

On Windows™ operating systems:

```
<DWC_home>\logs\dwcinstant<version_number>.log
```


On UNIX™ and Linux™ operating systems:

```
<DWC_DATA_dir>/installation/logs/dwcinst<version_number>.log
```

The log files for the agents are located in:

On Windows™ operating systems:

```
<TWA_home>\logs\twcinst<interp><user_name><version_number>.log
```

On UNIX™ and Linux™ operating systems:

```
<TWA_DATA_DIR>/installation/logs/twcinst<interp><user_name><version_number>.log
```

You can decide to rerun the installation or upgrade without executing the prerequisite scan. If you set the **-skipcheckprereq** parameter to `true` when performing the installation, the installation script does not execute the prerequisite scan. If a problem occurs, an error is displayed, the component is installed or upgraded, but might not work. For more information about the `-skipcheckprereq` parameter in all installation scripts, see the reference section in the *HCL Workload Automation: Planning and Installation*.

Connecting the Dynamic Workload Console to a new node or database

Move Dynamic Workload Console data to a new node or database by exporting data to an XML file to be imported in the new instance.

About this task

If you want to move the Dynamic Workload Console to a new node or database, you need to export the settings from an existing instance and create an XML file that can be imported into another Dynamic Workload Console node or database. If in your current environment you are using Derby, you can use this procedure to move to another supported database, because Derby is no longer supported starting from version 10.2.3.



Note: The migration of the roles from the Dynamic Workload Console Version 9.4 to Version 10.2.5 is not supported. You have to recreate the roles in the latest version.

To export the Dynamic Workload Console settings from the previous installation, perform the following procedure:

1. From the navigation toolbar, click **Administration > Manage Settings**.
2. In the Manage Settings panel, click **Export Settings** and save the XML file to a directory of your choice.
3. Optionally, edit the file using an XML editor and save it.
4. Optionally, export your custom boards: from the dashboard to be exported, click on the options menu next to the name of the dashboard and select **Export**. A JSON file is downloaded.
5. Browse to the `datasource_<db_vendor>.xml` file located in one of 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
```

6. Copy the `datasource_<db_vendor>.xml` to the path for your operating system:

On UNIX operating systems

`DWC_DATA_dir/usr/servers/dwcServer/configDropins/overrides`

On Windows operating systems

`DWC_home\usr\servers\dwcServer\configDropins\overrides`

7. Configure the `datasource_<db_vendor>.xml` file based on the specifics of your environment.

8. Only if you are moving from an Oracle database to a different database, browse to the following files:

- `DWC_DATA_dir>/usr/servers/dwcServer/apps/DWC.ear/DWCRest.war/META-INF/orm.xml`
- `DWC_DATA_dir>/usr/servers/dwcServer/apps/DWC.ear/Reporting.war/META-INF/orm.xml`
- `DWC_DATA_dir>/usr/servers/dwcServer/apps/DWC.ear/TWSWebUI.war/META-INF/orm.xml`

and replace the contents of the `<schema>` tag with `<schema>TDWC</schema>`.

9. Copy the settings file generated from the procedure to the workstation where the new Dynamic Workload Console is to be installed.

What to do next

You can now proceed with the upgrade, either direct or parallel, based on the procedure you have chosen. You will import the settings and boards when you install or upgrade the new Dynamic Workload Console.

If you need to connect the master domain manager to a new database, see the topic about connecting the master domain manager to a new database in *Administration Guide*.

Performing a direct upgrade from v 9.5.0.x or v 10.x.x to v 10.2.5

Detailed steps to perform a direct upgrade from version 9.5.0.x or v 10.x.x to version 10.2.5



To upgrade your environment using a direct upgrade procedure, perform the following steps:

1. [Upgrading WebSphere Application Server Liberty on page 239](#) on the workstations hosting Dynamic Workload Console and the server components (dynamic domain manager and its backups, master domain manager and its backups). You can also optionally move from WebSphere Application Server Liberty Base to Open Liberty.
2. [Performing a direct upgrade of the Dynamic Workload Console and its database on page 242](#)
3. [Performing a direct upgrade of the dynamic domain manager, its backups, and their database on page 245](#)
4. [Performing a direct upgrade of the backup master domain manager and its database on page 246](#)
 - a. [Switching the master domain manager to the upgraded backup master on page 249](#)
 - b. [Making the switch permanent on page 250](#)

5. [Performing a direct upgrade of the master domain manager on page 251](#)
 - a. [Switching back to the master domain manager from the backup master domain manager on page 253](#)
 - b. [Making the switch permanent on page 254](#)
6. [Upgrading agents and domain managers on page 255](#)

Environment with custom certificates

If you have version 9.5 installed with custom certificates, then after upgrading to 10.2 you must ensure that the parameters and the name of the relevant certificates in the **localopts** file are correct.

If you have previously used certificates generated with OpenSSL, check the paths in the following section:

- For Open SSL, check:
 - SSL key
 - SSL certified
 - SSL key pwd
 - SSL CA certified
 - SSL random seed

If you have used GSKit, the relevant parameters are automatically migrated to the new OpenSSL parameters:

- **SSL Version**
- **SSL Ciphers**
- **CLI SSL Ciphers**
- **CLI SSL Version**

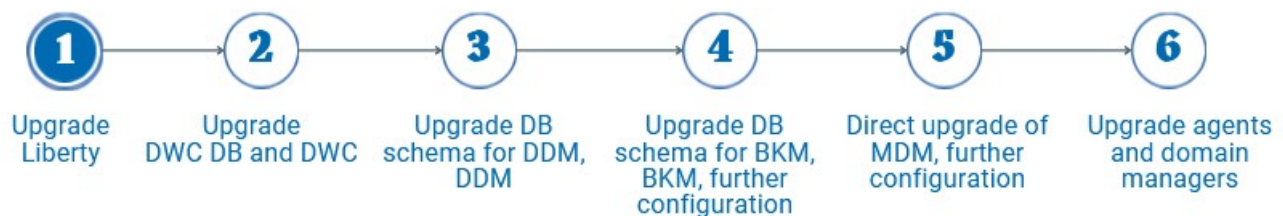
If the value of these fields corresponds to an incorrect path, then stop the WebSphere Application Server Liberty Base, make the necessary changes, and refer to the client's certificates, and then Restart.

For more information, see the section about setting local options in *Administration Guide*.

Upgrading WebSphere Application Server Liberty

You can optionally upgrade WebSphere Application Server Liberty to the latest supported version.

About this task



This is an optional step you might want to perform before you upgrade the Dynamic Workload Console and the master components. WebSphere Application Server Liberty refers both to WebSphere Application Server Liberty Base and Open Liberty.

If you have installed Open Liberty in your environment and you want to upgrade to the latest version, see [Upgrading Open Liberty on page 240](#).

If you have installed WebSphere Application Server Liberty Base in your environment and you want to upgrade to the latest version, see [Upgrading WebSphere Application Server Liberty Base on page 241](#).

If you want to move from WebSphere Application Server Liberty Base to Open Liberty, see the topic about moving from WebSphere Application Server Liberty Base to Open Liberty in *Administration Guide*.

Upgrading Open Liberty

About this task

Perform the following steps to upgrade Open Liberty to the latest supported version on the workstations hosting the Dynamic Workload Console and the server components (dynamic domain manager and its backups, master domain manager and its backups).

1. Find out which version of Open Liberty is required, by checking the required version of the Application server in the **Supported Software Report**, available in Product Requirements.
2. Download Open Liberty from [Get started with Open Liberty](#). Download the package named **All GA Features**
3. Stop the application server as described in the topic about application server - starting and stopping in *Administration Guide*.

Also stop HCL Workload Automation and all other applications running on the Open Liberty instance.

4. Optionally create a backup of the current Open Liberty instance in a directory different from the Open Liberty installation directory.
5. Uninstall Open Liberty.
6. Perform one of the following actions:
 - a. Extract Open Liberty using the root user:

On Windows operating systems

```
unzip <openliberty_download_dir>\openliberty-<version>.zip
-d <install_dir>
```

On UNIX operating systems

```
unzip <openliberty_download_dir>/openliberty-<version>.zip
-d <install_dir>
```

- b. Run the following command to assign permissions:

```
chmod 755 -R "wlp_directory"
```

OR

Extract Open Liberty using the user who is going to install the product, as follows:

```
su - "wuser"
unzip
```

where:

<openliberty_download_dir>

The directory where you downloaded Open Liberty.

install_dir

The directory where you want to install Open Liberty.



Note: Install the new Open Liberty in the exact location of the previous WebSphere Application Server Liberty Base installation.

7. Assign reading, writing, and processing permissions to libraries that are contained in the folder:

```
chmod -R 755 /open_liberty_installation_directory/wlp/lib/versions/
```

where:

open_liberty_installation_directory

The directory where you installed Open Liberty.

8. Restart the application server as described in Application server - starting and stopping.
Also restart HCL Workload Automation and all other applications running on the Open Liberty instance.

Upgrading WebSphere Application Server Liberty Base

About this task

Perform the following steps to upgrade WebSphere Application Server Liberty Base to the latest supported version on the workstations hosting the Dynamic Workload Console and the server components (dynamic domain manager and its backups, master domain manager and its backups).

1. Download WebSphere Application Server Liberty Base from .
Each WebSphere Application Server Liberty Base image is packaged as a . jar file named `wlp-base-all-version.jar`.

Check the release notes to ensure the latest WebSphere Application Server Liberty Base version is supported by HCL Workload Automation .
2. Stop the application server as described in the topic about application server - starting and stopping in *Administration Guide*.
Also stop HCL Workload Automation and all other applications running on the WebSphere Application Server Liberty Base instance.
3. Optionally create a backup of the current WebSphere Application Server Liberty Base instance in a directory different from the WebSphere Application Server Liberty Base installation directory.
4. Uninstall WebSphere Application Server Liberty Base.
5. Install WebSphere Application Server Liberty Base by extracting the archive file to a directory of your choice.

On Windows operating systems

```
java -jar liberty_download_dir\wlp-base-all-version.jar
--acceptLicense install_dir
```

On UNIX operating systems

```
java -jar liberty_download_dir/wlp-base-all-version.jar
--acceptLicense install_dir
```

where:

liberty_download_dir

The directory where you downloaded WebSphere Application Server Liberty Base.

install_dir

The directory where you want to upgrade WebSphere Application Server Liberty Base.



Note: Install the new WebSphere Application Server Liberty Base in the exact location of the previous WebSphere Application Server Liberty Base installation.

6. Restart the application server as described in the topic about application server - starting and stopping in *Administration Guide*.

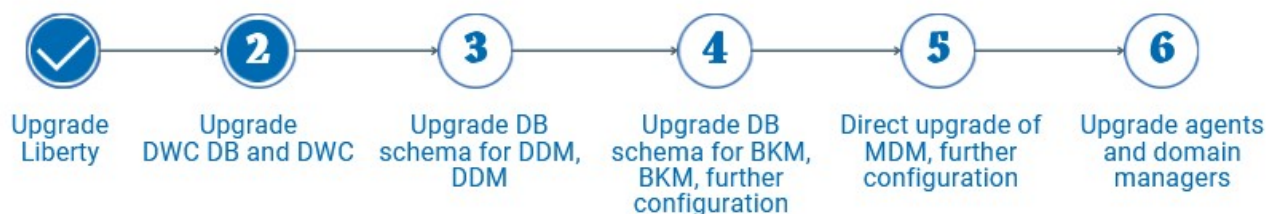
Also restart HCL Workload Automation and all other applications running on the WebSphere Application Server Liberty Base instance.

What to do next

You have now successfully upgraded WebSphere Application Server Liberty and can proceed to [Performing a direct upgrade of the Dynamic Workload Console and its database on page 242](#), to [Performing a direct upgrade of the backup master domain manager and its database on page 246](#), or to [Performing a direct upgrade of the master domain manager on page 251](#).

Performing a direct upgrade of the Dynamic Workload Console and its database

Perform a direct upgrade of the Dynamic Workload Console. If you have several Dynamic Workload Console nodes in a cluster, upgrade all the nodes in the cluster.

About this task

When upgrading the HCL Workload Automation environment, it is a good practice to update the Dynamic Workload Console first. If you update the console, you can then use it to verify that your environment is working after updating the remaining components.



Note: If you use Db2 for z/OS with the Dynamic Workload Console version 10.2.4 or later, transfer the drivers in binary mode from the directory where you installed Db2 for z/OS to a directory of your choice. When you run the `configuredb` or `dwcinst` script, set the directory you chose in the **dbdriverspath** parameter.



Note: If you are installing the Dynamic Workload Console version 10.2.3 or later, the Federator is also automatically installed. This component enables you to monitor your objects through the Orchestration Monitor page of the Dynamic Workload Console. For detailed information about how to configure and use the Federator, see [Mirroring the z/OS current plan to enable the Orchestration Monitor](#)the section about mirroring the z/OS current plan to enable the Orchestration Monitor in the *Dynamic Workload Console User's Guide*.



Note: If you are using a PostgreSQL database, check the collation settings before proceeding, as described in [Incorrect collation settings in PostgreSQL database on page 409](#).

If you are currently using Derby, you need to install a supported database and migrate your data. This is necessary because Derby is no longer supported as of version 10.2.3. For more information, see [Connecting the Dynamic Workload Console to a new node or database on page 237](#).

1. Log in to the workstation where you plan to install the Dynamic Workload Console.
2. On UNIX™ operating systems, ensure that **umask** is set to **022**. To verify that **umask** is set to the correct value, from a command prompt, run the **umask** command. If the value is different from **022**, modify it by running the following command:

```
umask 022
```

On Windows operating systems, ensure you have the correct rights on the folder where you plan to install.

3. Download the installation images from [HCL Software](#).
4. Browse to the folder *image_location*.
5. If possible, stop all Dynamic Workload Console instances.
If this is not possible, launch the `configureDB` script at a time when the Dynamic Workload Console is processing a low workload. If the `configureDB` script should fail because of conflicts with the Dynamic Workload Console, restart the script.
6. If your current version is earlier than 10.2.1, your certificates need to be updated before they can work with version 10.2.5. The update is performed automatically, but you need to provide the password for the certificates. Define an environment variable with name **JKS_SSL_PASSWORD** and set it to the password you defined for the certificates. You can optionally encrypt the password using the `secure` script. For more information about the `secure` script, see the [Reference](#) section.
7. To update the database version, run the following command:

On Windows operating systems

```
cscript configureDb.vbs --rdbmstype db_type --dbhostname db_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbpassword db_password --dbadminuser db_administrator
--dbadminuserpw db_administrator_password
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype db_type --dbhostname db_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbpassword db_password --dbadminuser db_administrator
--dbadminuserpw db_administrator_password
```

On z/OS operating systems

```
./configureDb.sh --rdbmstype db_type --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
--zlocationname zOS_location_containing_db --zbufferpoolname buffer_pool_in_zOS_location
```

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

8. Start the upgrade by launching the following command:

On Windows operating systems

```
cscript dwcinst.vbs --acceptlicense yes --inst_dir INST_DIR
```

On UNIX operating systems

```
./dwcinst.sh --acceptlicense yes --inst_dir INST_DIR
```

On z/OS operating systems

```
./dwcinst.sh --acceptlicense yes --inst_dir INST_DIR
```

For further details about commands, see [Reference on page 427](#).

9. If you had previously exported the Dynamic Workload Console, as described in [Connecting the Dynamic Workload Console to a new node or database on page 237](#), you can now import them in the new Dynamic Workload Console from the **Administration > Manage Settings** menu. If you have a high availability configuration, import the settings on one node.
10. If you have copied any template `.xml` files from the `templates` folder to the `overrides` folder, check for any differences between the default `.xml` files just upgraded in the `templates` folder and the files you are using in the `overrides` folder. If any differences are present, update the files in the `overrides` folder accordingly.

What to do next

You have now successfully upgraded the Dynamic Workload Console. You can now proceed to upgrade domain managers using the procedure described in [Performing a direct upgrade of the dynamic domain manager, its backups, and their database on page 245](#).

Performing a direct upgrade of the dynamic domain manager, its backups, and their database

Complete this procedure to upgrade the dynamic domain manager and the backup dynamic domain manager.

About this task



Upgrade a dynamic domain manager and a backup dynamic domain manager from version 9.5.0.x to version 10.2.x by running the **serverinst** script. Launch the script on the workstation where the dynamic domain manager is running to upgrade the dynamic domain manager, then launch the script on the workstation where the backup dynamic domain manager is running to upgrade the backup dynamic domain manager.



Note: If you are using a PostgreSQL database, check the collation settings before proceeding, as described in [Incorrect collation settings in PostgreSQL database on page 409](#).

1. Log in to the workstation where you plan to install.
2. On UNIX™ operating systems, ensure that **umask** is set to **022**. To verify that **umask** is set to the correct value, from a command prompt, run the **umask** command. If the value is different from **022**, modify it by running the following command:

```
umask 022
```

On Windows operating systems, ensure you have the correct rights on the folder where you plan to install.

3. Download the installation images from [HCL Software](#).
4. Browse to the folder `image_location/TWS/interp_name`.
5. Stop all HCL Workload Automation services and WebSphere Application Server Liberty, by running the following commands:

```
conman stop; wait
conman shut; wait
conman ShutDownLwa
stopappserver
```

6. To update the database version, run the following command:

On Windows operating systems

```
cscript configureDb.vbs --componenttype=DDM --rdbmstype db_type --dbhostname db_hostname
--dbport db_port --dbname db_name --dbuser db_user --dbpassword db_password --dbadminuser
db_administrator --dbadminuserpw db_administrator_password
```

On UNIX operating systems

```
./configureDb.sh --componenttype=DDM --rdmstype db_type --dbhostname db_hostname --
dbport db_port --dbname db_name --dbuser db_user --dbpassword db_password --dbadminuser
db_administrator --dbadminuserpw db_administrator_password
```

For more information about the **configureDb** script, see [Database configuration - configureDb script on page 430](#).

7. If your current version is earlier than 10.2.1, your certificates need to be updated before they can work with version 10.2.5. The update is performed automatically, but you need to provide the password for the certificates. Define an environment variable with name **JKS_SSL_PASSWORD** and set it to the password you defined for the certificates. You can optionally encrypt the password using the secure script. For more information about the secure script, see the Reference section.
8. Check your FIPS settings. If FIPS is not enabled in your current environment, you can skip the **enablefips** parameter.
9. Start the installation launching the following command:

On Windows operating systems

```
cscript serverinst.vbs --acceptlicense yes --inst_dir INST_DIR
```

On UNIX operating systems

```
./serverinst.sh --acceptlicense yes --inst_dir INST_DIR
```

where *INST_DIR* is the directory where the component is installed. To find out the installation directory, see the topic about finding out what has been installed in which HCL Workload Automation instances in *Administration Guide*.



Note: The **acceptlicense** and **inst_dir** parameters are required. All other parameters are ignored by the **serverinst** command, except for the following two optional parameters: **lang** and **skipcheckprereq**.

For further details about the **serverinst** script, see [Server components installation - serverinst script on page 442](#).

10. If you have copied any template `.xml` files from the `templates` folder to the `overrides` folder, check for any differences between the default `.xml` files just upgraded in the `templates` folder and the files you are using in the `overrides` folder. If any differences are present, update the files in the `overrides` folder accordingly.
11. After the installation has completed, run the following commands to start up HCL Workload Automation services and WebSphere Application Server Liberty:

```
conman start
conman startappserver
StartUpLwa
```

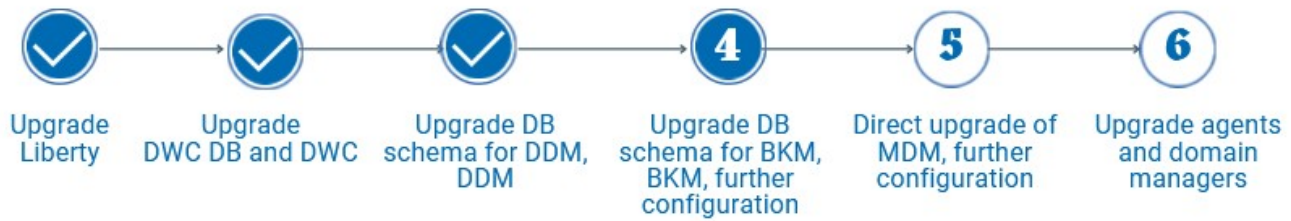
What to do next

You have now successfully upgraded the dynamic domain manager and its backup. You can now proceed to [Performing a direct upgrade of the backup master domain manager and its database on page 246](#).

Performing a direct upgrade of the backup master domain manager and its database

Performing a direct upgrade of the backup master domain manager.

About this task



Upgrade a backup master domain manager and its database.



Note: If you are using a PostgreSQL database, check the collation settings before proceeding, as described in [Incorrect collation settings in PostgreSQL database on page 409](#).

1. Log in to the workstation where you plan to install.
2. On UNIX™ operating systems, ensure that **umask** is set to **022**. To verify that **umask** is set to the correct value, from a command prompt, run the **umask** command. If the value is different from **022**, modify it by running the following command:

```
umask 022
```

On Windows operating systems, ensure you have the correct rights on the folder where you plan to install.

3. Download the installation images from [HCL Software](#).
4. Browse to the folder `<image_location>/TWS/interp_name`.
5. Stop all HCL Workload Automation services and WebSphere Application Server Liberty, by running the following commands:

```
conman "stop;wait"
conman "stopappserver;wait"
conman "shut;wait"
ShutDownLwa
```

6. If your current version is earlier than 10.2.1, your certificates need to be updated before they can work with version 10.2.5. The update is performed automatically, but you need to provide the password for the certificates. Define an environment variable with name **JKS_SSL_PASSWORD** and set it to the password you defined for the certificates. You can optionally encrypt the password using the secure script. For more information about the secure script, see the Reference section.
7. To update the database version, run the following command:

On Windows operating systems

```
cscript configureDb.vbs --rdbmstype db_type --dbhostname db_hostname --dbport db_port --
dbname db_name --dbuser db_user --dbpassword db_password --dbadminuser db_administrator --
dbadminuserpw db_administrator_password
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype db_type --dbhostname db_hostname --dbport db_port --dbname db_name
--dbuser db_user --dbpassword db_password --dbadminuser db_administrator --dbadminuserpw
db_administrator_password
```

For more information about the configureDb script, see [Database configuration - configureDb script on page 430](#).

8. Check your FIPS settings. If FIPS is not enabled in your current environment, you can skip the **enablefips** parameter.
9. Start the installation launching the following command:

On Windows operating systems

```
cscript serverinst.vbs --acceptlicense yes --inst_dir INST_DIR
```

On UNIX operating systems

```
./serverinst.sh --acceptlicense yes --inst_dir INST_DIR
```

where *INST_DIR* is the directory where the component is installed. To find out the installation directory, see the topic about finding out what has been installed in which HCL Workload Automation instances in *Administration Guide*.

For more information about the serverinst script, see [Server components installation - serverinst script on page 442](#).



Note: The **acceptlicense** and **inst_dir** parameters are required. You can also specify the following optional parameters:

- **lang**
- **work_dir**
- **skipcheckprereq**
- **enablefips**

If you specify other parameters, they are ignored and the settings from the current instance are used instead.

10. After the installation has completed, run the following commands to start up HCL Workload Automation services and WebSphere Application Server Liberty:

```
conman start
conman startappserver
StartUpLwa
```

11. To link all fault-tolerant agents, type the following command:

```
conman "link @!/!/@"
```

12. If you have copied any template `.xml` files from the `templates` folder to the `overrides` folder, check for any differences between the default `.xml` files just upgraded in the `templates` folder and the files you are using in the `overrides` folder. If any differences are present, update the files in the `overrides` folder accordingly.

What to do next

You have now successfully upgraded the backup master domain manager. You can now proceed to [Switching the master domain manager to the upgraded backup master on page 249](#).

Switching the master domain manager to the upgraded backup master

About this task



To switch the back-level master domain manager to the upgraded backup master domain manager, complete the following procedure:

1. Switch to your upgraded backup master domain manager, which now becomes your current active master domain manager, by issuing the following command from either the Dynamic Workload Console or the **command line** of your back-level master domain manager:

From the Dynamic Workload Console

In the navigation tree, click **Monitoring and Reporting > Monitor Workload >** select the engine and the object type Workstation, click run and, in the table of results, select backup master domain manager workstation name, click **More Actions**, and select **Become Master Domain Manager**.

From the command line of the back-level master domain manager

Issue the following command:

```
conman "switchmgr masterdm;new_mgr_cpu"
```

where *new_mgr_cpu* is the backup master domain manager workstation name.

2. Switch the event processor from the back-level master domain manager to the backup master domain manager, by running the following command from either the Dynamic Workload Console or the **command line** of your back-level master domain manager:

From the Dynamic Workload Console

In the navigation tree, click **Monitoring and Reporting > Monitor Workload >** select the engine and the object type Workstation, click run and, in the table of results, select backup master domain manager workstation name, click **More Actions**, and select **Become Event Processor**.

From the command line of the back-level master domain manager

Issue the following command:

```
conman "switcheventprocessor new_mgr_cpu"
```

where *new_mgr_cpu* is the backup master domain manager workstation name.

Results

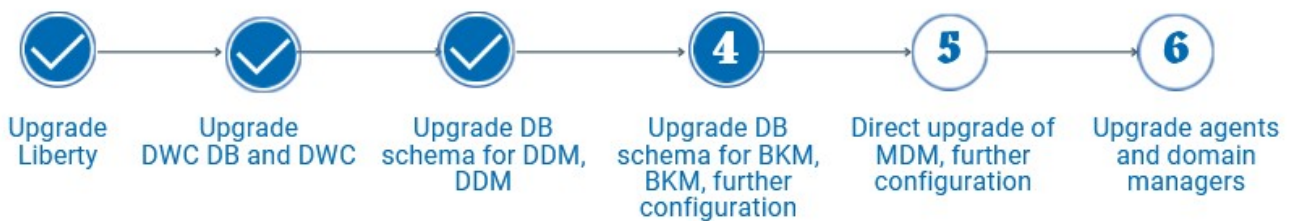
Once you have switched the master domain manager to the upgraded backup master, you can make this switch permanent. For details, see [Making the switch permanent on page 250](#).

For more detailed information about switching the master domain manager, see the related topic in the *Administration Guide*

Making the switch permanent

Making the switch manager permanent

About this task



In the procedure [Switching the master domain manager to the upgraded backup master on page 249](#), you switched your master domain manager promoting your new version backup master domain manager to the role of master domain manager.

To make this configuration fully operational and persistent through **JnextPlan**, you must complete the following procedure:

What to do next

To make this configuration fully operational and persistent through **JnextPlan**, complete the following procedure on the new manager, referred to as *new_mgr_cpu*:

1. Edit the *localopts* file and modify the following entry as shown:

```
DEFAULTWS=new_mgr_cpu
```

where *new_mgr_cpu* is the workstation name of the new manager. For more information about *localopts* file, see the section about setting local options in *Administration Guide*.

2. Change the workstation definition of the back-level master by running:

```
composer modify cpu=old_mgr_cpu
```

and in the definition substitute *type=manager* with *type=fta*

3. Change the workstation definition of the new manager by running:

```
composer modify cpu=new_mgr_cpu
```

and in the definition substitute *type=fta* with *type=manager*.

4. Ensure that the **optman cf** option is set to *all*.

5. Rebuild the plan to activate the changes to the database:

```
JnextPlan -for 0000
```

6. Restore the previous setting of the **optman** `cf` option, if necessary.
7. Edit the `/TWA_DATA_DIR/mozart/globalopts` file and modify the **master=old_mgr_cpu** entry as shown:

```
master=new_mgr_cpu
```

where *new_mgr_cpu* is the workstation name of the new master. For more information about **optman**, see the section about setting global options in *Administration Guide*.

In this way the reports `reptr-pre` and `reptr-post` can run when you run **JnextPlan**.

Once you have made the switch manager permanent, you must run the FINAL job stream on the new master domain manager.

You can now proceed to [Performing a direct upgrade of the master domain manager on page 251](#).

Performing a direct upgrade of the master domain manager

Performing a direct upgrade of the master domain manager

About this task



Upgrade a master domain manager by running the **serverinst** script.



Note: If you are using a PostgreSQL database, check the collation settings before proceeding, as described in [Incorrect collation settings in PostgreSQL database on page 409](#).

1. Log in to the workstation where you plan to install.
2. On UNIX™ operating systems, ensure that **umask** is set to **022**. To verify that **umask** is set to the correct value, from a command prompt, run the **umask** command. If the value is different from **022**, modify it by running the following command:

```
umask 022
```

On Windows operating systems, ensure you have the correct rights on the folder where you plan to install.

3. Download the installation images from [HCL Software](#).
4. Browse to the folder `<image_location>/TWS/interp_name`.
5. Stop all HCL Workload Automation services and WebSphere Application Server Liberty, by running the following commands:

```
conman "stop;wait"
conman "stopappserver;wait"
conman "shut;wait"
ShutDownLwa
```

6. If your current version is earlier than 10.2.1, your certificates need to be updated before they can work with version 10.2.5. The update is performed automatically, but you need to provide the password for the certificates. Define an environment variable with name **JKS_SSL_PASSWORD** and set it to the password you defined for the certificates. You can optionally encrypt the password using the secure script. For more information about the secure script, see the Reference section.
7. To update the database version, run the following command:

On Windows operating systems

```
cscript configureDb.vbs --rdbmstype db_type --dbhostname db_hostname --dbport db_port --
dbname db_name --dbuser db_user --dbpassword db_password --dbadminuser db_administrator --
dbadminuserpw db_administrator_password
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype db_type --dbhostname db_hostname --dbport db_port --dbname db_name
--dbuser db_user --dbpassword db_password --dbadminuser db_administrator --dbadminuserpw
db_administrator_password
```

For more information about the configureDb script, see [Database configuration - configureDb script on page 430](#).

8. Check your FIPS settings. If FIPS is not enabled in your current environment, you can skip the **enablefips** parameter.
9. Start the installation launching the following command:

On Windows operating systems

```
cscript serverinst.vbs --acceptlicense yes --inst_dir INST_DIR
```

On UNIX operating systems

```
./serverinst.sh --acceptlicense yes --inst_dir INST_DIR
```

where *INST_DIR* is the directory where the component is installed. To find out the installation directory, see the topic about finding out what has been installed in which HCL Workload Automation instances in *Administration Guide*.

For more information about the serverinst script, see [Server components installation - serverinst script on page 442](#).



Note: The **acceptlicense** and **inst_dir** parameters are required. You can also specify the following optional parameters:

- **lang**
- **work_dir**
- **skipcheckprereq**
- **enablefips**



If you specify other parameters, they are ignored and the settings from the current instance are used instead.

- After the installation has completed, run the following commands to start up HCL Workload Automation services and WebSphere Application Server Liberty:

```
conman start
conman startappserver
StartUpLwa
```

- To link all fault-tolerant agents, type the following command:

```
conman "link @!/@/@ "
```

- If you have copied any template .xml files from the `templates` folder to the `overrides` folder, check for any differences between the default .xml files just upgraded in the `templates` folder and the files you are using in the `overrides` folder. If any differences are present, update the files in the `overrides` folder accordingly.

What to do next

You have now successfully upgraded the master domain manager. You can now proceed to [Switching back to the master domain manager from the backup master domain manager on page 253](#).

Switching back to the master domain manager from the backup master domain manager

About this task



After upgrading the old master domain manager to the 10.2.x version, you can now switch back the master capabilities, so that you restore your environment to the previous state, as follows:

- Stop the application server as described in the topic about application server - starting and stopping in *Administration Guide*.
- Switch the upgraded backup master domain manager, which now becomes the master domain manager, by issuing the following command from either the Dynamic Workload Console or the **command line** of your current backup master domain manager:

From the Dynamic Workload Console

In the navigation tree, click **Monitoring and Reporting > Monitor Workload >** select the engine and the object type Workstation, click run and, in the table of results, select backup master domain manager workstation name, click **More Actions**, and select **Become Master Domain Manager**.

From the command line of the back-level master domain manager

Issue the following command:

```
conman "switchmgr masterdm;new_mgr_cpu"
```

where *new_mgr_cpu* is the backup master domain manager workstation name.

3. Switch the event processor from the old backup master domain manager to the master domain manager, by running the following command from either the Dynamic Workload Console or the **command line** of your old backup master domain manager:
4. Restart the application server as described in the topic about application server - starting and stopping in *Administration Guide*.

What to do next

You have now successfully switched back the upgraded master domain manager. You can now proceed to [Making the switch permanent on page 254](#).

Making the switch permanent

Making the switch manager permanent

About this task



In the procedure [Switching the master domain manager to the upgraded backup master on page 249](#), you switched your master domain manager promoting your new version backup master domain manager to the role of master domain manager.

To make this configuration fully operational and persistent through **JnextPlan**, you must complete the following procedure:

To make this configuration fully operational and persistent through **JnextPlan**, complete the following procedure on the new manager, referred to as *new_mgr_cpu*:

1. Edit the *localopts* file and modify the following entry as shown:

```
DEFAULTWS=new_mgr_cpu
```

where *new_mgr_cpu* is the workstation name of the new manager. For more information about *localopts* file, see the section about setting local options in *Administration Guide*.

2. Change the workstation definition of the back-level master by running:

```
composer modify cpu=old_mgr_cpu
```

and in the definition substitute *type=manager* with *type=fta*

3. Change the workstation definition of the new manager by running:

```
composer modify cpu=new_mgr_cpu
```

and in the definition substitute *type=fta* with *type=manager*.

4. Ensure that the **optman** cf option is set to *all*.
5. Rebuild the plan to activate the changes to the database:

```
JnextPlan -for 0000
```

6. Restore the previous setting of the **optman** cf option, if necessary.
7. Edit the `/TWA_DATA_DIR/mozart/globalopts` file and modify the **master=old_mgr_cpu** entry as shown:

```
master=new_mgr_cpu
```

where *new_mgr_cpu* is the workstation name of the new master. For more information about optman, see the section about setting global options in *Administration Guide*.

In this way the reports `reptr-pre` and `reptr-post` can run when you run **JnextPlan**.

What to do next

Once you have made the switch manager permanent, you must run the FINAL job stream on the new master domain manager.

You can now proceed to [Upgrading agents and domain managers on page 255](#).

Upgrading agents and domain managers

There are several methods you can choose from to upgrade your domain managers and agents.



The agent upgrade can be performed with minimal impact to scheduling activities. The agents are stopped for the shortest time necessary to perform the maintenance. Any active agent command-line interfaces and processes, such as `conman`, `composer`, `netman`, `mailman`, and `batchman`, to name a few, continue running. Any jobs already running when the upgrade process begins, continue to run as planned, however, no new jobs begin execution during this time. Once the upgrade is complete, the agent is restarted and quickly reconnects with its jobs. Any jobs that were actively running before the upgrade that have not yet completed, continue to run, and any jobs that successfully finished running during the upgrade procedure report a successful job status. An automatic backup and restore feature is in place in case of failure.

Because domain managers are agents, they are upgraded using the procedures described in this section.

If you choose to upgrade your environment top-down, then the agents get upgraded progressively after you have upgraded the master domain manager and its backup. This means that new features and enhancements are not available on all of your agents at the same time. If, instead, you choose to upgrade your environment bottom-up, then the agents are upgraded first, and new features and enhancements become available after the master domain manager and its backup have been upgraded.



Important: After upgrading your fault-tolerant agents, it might be necessary to 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. See the topic about updating the security file in the *Administration Guide* for more information.

You can choose to upgrade your agents using any of the following methods:

twinsinst script

A single line command that checks if processes or a command line is running before it starts. It saves disk space and RAM because it is not Java-based. See [Upgrade procedure on page 290](#) and [Upgrading agents on IBM i systems on page 295](#)

Centralized agent update

Upgrade or update multiple fault-tolerant agent and dynamic agent instances at the same time. Download the fix pack installation package, or the elmage upgrade package to the master domain manager and then either run the installation on multiple agent instances or schedule the installation by creating and submitting a job to run. This upgrade method is not supported on z-centric agent instances. See [Centralized agent update on page 300](#).

For a list of supported operating systems and requirements, see the System Requirements Document at [HCL Workload Automation Detailed System Requirements](#).

When the upgrade procedure has completed successfully, the backup instance is deleted.



Note: 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 following path:

On Windows operating systems

```
<TWA_home>\TWS
```

On UNIX operating systems

```
<TWA_DATA_DIR>
```

When upgrading dynamic agents featuring both a local and a remote gateway, ensure you either upgrade the agent first and then the gateway or upgrade both at the same time.

Parallel upgrade from version 9.5.0.x or 10.x.x to version 10.2.5

About this task



To upgrade your environment using a parallel upgrade procedure, perform the following steps:

1. [Converting default certificates on page 258](#), if you are using default certificates in your current environment. Use this procedure to convert the certificates from the JKS to the PEM format, then copy them to the workstations where you plan to install the server components (dynamic domain manager and its backups, master domain manager and its backups) and the Dynamic Workload Console.

Certificates are now required when installing or upgrading HCL Workload Automation. You can no longer install nor upgrade HCL Workload Automation without securing your environment with certificates. The required certificates are:

- ca.crt
- tls.key
- tls.crt

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

2. [Upgrading WebSphere Application Server Liberty on page 260](#)
3. [Encrypting passwords \(optional\) on page 262](#)
4. [Upgrading the Dynamic Workload Console and its database on page 264](#)
5. [Creating the HCL Workload Automation administrative user on page 266](#) on the workstations which will host the components at 10.2.5 level.
6. [Upgrading the database for the server components on page 268](#)
7. [Installing a new dynamic domain manager configured as a backup on page 271](#)
 - a. [Switching the manager to the upgraded backup on page 273](#)
 - b. [Making the switch permanent on page 274](#)
8. [Installing the new master domain manager configured as a backup on page 276](#)
 - a. [Switching the manager to the upgraded backup on page 281](#)
 - b. [Making the switch permanent on page 282](#)
9. [Customizing and submitting the optional FINAL job stream on page 283](#)
10. [Installing a new backup dynamic domain manager on page 285](#) to replace the backup dynamic domain manager which you have switched to become the current dynamic domain manager.
11. [Cleaning up your environment on page 288](#)
12. [Optionally dismiss all back-level components](#)
13. [Upgrading agents and domain managers on page 289](#)
14. [Optionally install a new backup master domain manager at version 10.2.5 to ensure failover capabilities.](#)

Environment with custom certificates

If you have version 9.5 installed with custom certificates, then after upgrading to 10.2 you must ensure that the parameters and the name of the relevant certificates in the **localopts** file are correct.

If you have previously used certificates generated with OpenSSL, check the paths in the following section:

- For Open SSL, check:
 - SSL key
 - SSL certified
 - SSL key pwd
 - SSL CA certified
 - SSL random seed

If you have used GSKit, the relevant parameters are automatically migrated to the new OpenSSL parameters:

- **SSL Version**
- **SSL Ciphers**
- **CLI SSL Ciphers**
- **CLI SSL Version**

Converting default certificates

Procedure to extract and convert default certificates generated in your current version prior to upgrading.

About this task



If you are using default certificates, extract and convert them before you start the upgrade. Perform the following steps:

1. Set the HCL Workload Automation environment, as described in [Setting the environment variables on page 206](#).
2. To ensure the keytool and openssl commands start correctly on all operating systems, browse to the folder where the keytool and openssl commands are located and launch the commands as follows:

```
cd <TWS_DIR>/JavaExt/jre/jre/bin

./keytool -importkeystore -srckeystore TWSServerKeyFile.jks -destkeystore
<path_of_extracted_certs>/server.p12 -deststoretype pkcs12

cd <TWS_DIR>/tmpOpenSSL64/1.1/bin/openssl

./openssl pkcs12 -in <path_of_extracted_certs>/server.p12 -out
<path_of_extracted_certs>/tls.tot
```

The location of the `TWSServerKeyFile.jks` varies depending on the HCL Workload Automation version you have currently installed, as follows:

versions 9.5 and later

```
TWA_DATA_DIR/usr/servers/engineServer/resources/security
```

versions 9.4 and earlier

```
TWA_home/WAS/TWSPProfile/etc
```

3. Open the `tls.tot` file with any text editor.
4. From the `tls.tot` file, copy the private key to a new file named `tls.key`.

The `tls.key` file must be structured as follows:

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
<private_key>
-----END ENCRYPTED PRIVATE KEY-----
```



Note: Insert a carriage return after each key, so that an empty line is inserted after each key.

5. From the `tls.tot` file, copy the public key to a new file named `tls.crt`.

The `tls.crt` file must be structured as follows:

```
-----BEGIN CERTIFICATE-----
<public_key>
-----END CERTIFICATE-----
```



Note: Insert a carriage return after each key, so that an empty line is inserted after each key.

6. Copy the contents of the `tls.crt` file into a new file named `ca.crt`. If you want to upgrade a dynamic domain manager, also copy the contents of the `tls.crt` file into another new file named `jwt.crt`.
7. Create a file named `tls.sth` containing the passphrase you have specified for creating the `.p12` certificate in step 2 on page 258, encoded in `base64` format. To create the `tls.sth` file, use the following command:

```
./secure -password your_password -base64 e -out
<path_of_extracted_certs>/tls.sth
```

If you are using a version earlier than 10.x, you can find the `secure` script in the installation package of the 10.2.5 version you are upgrading to. You can launch the script from one of the following paths:

master domain manager and agent

```
<10.2.5_extracted_image_dir>/TWS/<interp>/Tivoli_LWA-<interp>/TWS/bin
```

Dynamic Workload Console

```
<10.2.5_extracted_image_dir>/DWC/<interp>/bin
```

where

<interp>

is the operating system you are installing on

As an alternative, you can use the following command on UNIX workstations:

```
echo -n "passwordToEncode" | base64 >> tls.sth
```

8. Browse to the GSKit folder and extract the client certificates from the `TWA_DATA_DIR/ssl/GSKit` folder by running the following commands, depending on the HCL Workload Automation version you have currently installed:

```
cd <TWS_DIR>/tmpGSKit64/8/bin
```

versions 9.5 and later

```
./gsk8capi64 -cert -extract -db <TWA_DATA_DIR>/ssl/GSKit/TWSClientKeyStore.kdb  
-stashed -label client -target <path_of_extracted_certs>/client.crt
```

versions 9.4 and earlier

```
./gsk8capi64 -cert -extract -db <TWS_DIR>/ssl/GSKit/TWSClientKeyStore.kdb  
-stashed -label client -target <path_of_extracted_certs>/client.crt
```

9. Create a folder named `additionalCAs` in the folder where you extracted the certificates and move the `client.crt` file created in step 8 on page 260 to the `additionalCAs` folder.
10. Insert the `client.crt` in the `additionalCAs` folder when providing the certificates to the installation script with the **sslkeysfolder** parameter.
11. Assign the correct permissions (755) and ownerships to extracted certificates, as follows:

```
chmod -R 755 <path_of_extracted_certs>
```

Results

You have now extracted and converted your certificates for use with version 10.2.5.

What to do next

You can now upgrade WebSphere Application Server Liberty, as described in [Upgrading WebSphere Application Server Liberty on page 260](#). When upgrading HCL Workload Automation components in upcoming steps, provide the path to the folder where you extracted the certificates using the **sslkeysfolder** parameter when running the installation scripts. For more information about the installation scripts, see [Reference on page 427](#).

Upgrading WebSphere Application Server Liberty

Upgrading WebSphere Application Server Liberty to the latest supported version. This is an optional step you might want to perform before you upgrade the Dynamic Workload Console and the server components.



Before you begin

On AIX and Linux workstations, ensure you permanently set the **ulimit** parameter as follows:

- data segment process (option **-d**) = unlimited
- file size (option **-f**) = unlimited
- max user processes (option **-u**) = >260000 up to unlimited
- open files (option **-n**) = >100000 up to unlimited

- max memory size (option **-m**) = `unlimited`
- stack size (option **-s**) = `>33000 up to unlimited`

On the master domain manager, these settings must be applied to:

- `root`
- the HCL Workload Automation administrative user

On the Dynamic Workload Console, these settings must be applied to:

- `root`
- the Dynamic Workload Console installation user (if this user is different from root)

Ensure that your system meets the operating system requirements. For more information, see Open Liberty detailed system requirements.

About this task

You can quickly install Open Liberty by extracting an archive file on all supported platforms.

If you already have WebSphere Application Server Liberty Base installed, you can use it with HCL Workload Automation, otherwise you can install Open Liberty, as described below.

If you want to move from WebSphere Application Server Liberty Base to Open Liberty, see the topic about moving from WebSphere Application Server Liberty Base to Open Liberty in *Administration Guide*.

Install Open Liberty on all of the following workstations, which comprise a typical installation:

- master domain manager
- backup domain manager
- two Dynamic Workload Console installations on two separate workstations

If you plan to install a dynamic domain manager and its backup, these components require a separate Open Liberty installation.

On UNIX workstations, you can install Open Liberty using a user of your choice. In this case, assign the HCL Workload Automation administrative user read and write access to the Open Liberty installation directory.

To install Open Liberty, perform the following steps:

1. Find out which version of Open Liberty is required, by checking the required version of the Application server in the **Supported Software Report**, available in Product Requirements.
2. Download Open Liberty from [Get started with Open Liberty](#). Download the package named **All GA Features**
3. Perform one of the following actions:

- a. Extract Open Liberty using the root user:

On Windows operating systems

```
unzip <openliberty_download_dir>\openliberty-<version>.zip
-d <install_dir>
```

On UNIX operating systems

```
unzip <openliberty_download_dir>/openliberty-<version>.zip
-d <install_dir>
```

- b. Run the following command to assign permissions:

```
chmod 755 -R "wlp_directory"
```

OR

Extract Open Liberty using the user who is going to install the product, as follows:

```
su - "wuser"
unzip
```

where:

<openliberty_download_dir>

The directory where you downloaded Open Liberty.

install_dir

The directory where you want to install Open Liberty.



Note: Install the new Open Liberty in the exact location of the previous WebSphere Application Server Liberty Base installation.

4. Ensure the HCL Workload Automation administrative user has the rights to run Open Liberty and full access to the installation directory. If Open Liberty is shared between the master domain manager and the Dynamic Workload Console, ensure also the Dynamic Workload Console user has the same rights.

Results

You have now successfully installed Open Liberty.

What to do next

You can now proceed to [Encrypting passwords \(optional\) on page 262](#) or to [Upgrading the Dynamic Workload Console and its database on page 264](#).

Encrypting passwords (optional)

How to encrypt passwords required by the installation process

About this task



You can optionally encrypt the passwords that you will use while installing, upgrading, and managing HCL Workload Automation. The secure command uses the AES method and prints the encrypted password to the screen or saves it to a file.



Note: It is important you understand the limits to the protection that this method provides. The custom passphrase you use to encrypt the passwords is stored in clear format in the `passphrase_variables.xml` file, stored in `configureDropin`. To fully understand the implications of this method, it is recommended you read the information provided by Open Liberty at the link [Password encryption limitations](#).

You can perform a typical procedure, which uses a custom passphrase, as described in the following scenario. For more information about all secure arguments and default values, see [Optional password encryption - secure script on page 427](#).

Encrypting the password

1. Browse to the folder where the secure command is located:
 - Before the installation, the command is located in the product image directory, `<image_directory>/TWS/<op_sys>/Tivoli_LWA_<op_sys>/TWS/bin`
 - After the installation, the command is located in `TWA_home/TWS/bin`
2. Depending on your operating system, encrypt the password as follows:

Windows operating systems

```
secure -password password -passphrase passphrase
```

UNIX operating systems

```
./secure -password password -passphrase passphrase
```

z/OS operating systems

```
./secure -password password -passphrase passphrase
```

where

-password

Specifies the password to be encrypted.

-passphrase

Specifies the custom passphrase that is used to generate the key with which the command encrypts the password. If you set this parameter, inform the user who installs HCL Workload Automation that they must define the **SECUREWRAP_PASSPHRASE** environment variable in the same shell from which they run the installation command, and set it to the same value as the **passphrase** parameter. On Windows operating systems, the passphrase must be at least 8 characters long. This argument generates a password which can be reused for all HCL Workload Automation components. This parameter is

mutually exclusive with the [-useaeskeystore on page 429](#) parameter, which generates a password which can be decrypted only on the local workstation and not reused for other components.

3. Provide both the encrypted password and custom passphrase to the user in charge of installing HCL Workload Automation. You can use encrypted passwords only in association with the specific passphrase used to encrypt them.

Installing with the encrypted password

The user in charge of installing HCL Workload Automation must set the **SECUREWRAP_PASSPHRASE** environment variable by performing the following steps:

1. Open a brand new shell session.
2. Ensure that no value is set for the **SECUREWRAP_PASSPHRASE** environment variable.
3. Define the **SECUREWRAP_PASSPHRASE** environment variable and set it to the passphrase defined by the user who ran the secure command, as follows:

```
SECUREWRAP_PASSPHRASE=<passphrase>
```

You can use encrypted passwords only in association with the specific passphrase used to encrypt them.

4. In the same shell session, provide the encrypted passwords when running any command that uses a password. An encrypted password looks like the following example:

```
{aes}AFC3jj9cR0YyqR+3CONBzVi8deLb2Bossb9GGroh8UmDPGikIkzXZzid3nzY0IhnSg=
```

What to do next

You can now proceed to [Upgrading the Dynamic Workload Console and its database on page 264](#).

Upgrading the Dynamic Workload Console and its database

Upgrade the Dynamic Workload Console from version 9.5.0.x or 10.x to version 10.2.x. If you have several Dynamic Workload Console nodes in a cluster, upgrade all the nodes in the cluster.

About this task



When upgrading the HCL Workload Automation environment, it is a good practice to update the Dynamic Workload Console first. If you update the console, you can then use it to verify that your environment is working after updating the remaining components.



Note: If you use Db2 for z/OS with the Dynamic Workload Console version 10.2.4 or later, transfer the drivers in binary mode from the directory where you installed Db2 for z/OS to a directory of your choice. When you run the `configuredb` or `dwcinst` script, set the directory you chose in the **dbdriverspath** parameter.



Note: If you are installing the Dynamic Workload Console version 10.2.3 or later, the Federator is also automatically installed. This component enables you to monitor your objects through the Orchestration Monitor page of the Dynamic Workload Console. For detailed information about how to configure and use the Federator, see [Mirroring the z/OS current plan to enable the Orchestration Monitor](#) the section about mirroring the z/OS current plan to enable the Orchestration Monitor in the *Dynamic Workload Console User's Guide*.



Note: If you are using a PostgreSQL database, check the collation settings before proceeding, as described in [Incorrect collation settings in PostgreSQL database on page 409](#).

If you are currently using Derby, you need to install a supported database and migrate your data. This is necessary because Derby is no longer supported as of version 10.2.3. For more information, see [Connecting the Dynamic Workload Console to a new node or database on page 237](#).

1. Log in to the workstation where you plan to install the Dynamic Workload Console.
2. On UNIX™ operating systems, ensure that **umask** is set to **022**. To verify that **umask** is set to the correct value, from a command prompt, run the **umask** command. If the value is different from **022**, modify it by running the following command:

```
umask 022
```

On Windows operating systems, ensure you have the correct rights on the folder where you plan to install.

3. Download the installation images from [HCL Software](#).
4. Browse to the folder *image_location*.
5. If possible, stop all Dynamic Workload Console instances.
If this is not possible, launch the `configureDB` script at a time when the Dynamic Workload Console is processing a low workload. If the `configureDB` script should fail because of conflicts with the Dynamic Workload Console, restart the script.
6. If your current version is earlier than 10.2.1, your certificates need to be updated before they can work with version 10.2.5. The update is performed automatically, but you need to provide the password for the certificates. Define an environment variable with name **JKS_SSL_PASSWORD** and set it to the password you defined for the certificates. You can optionally encrypt the password using the `secure` script. For more information about the `secure` script, see the [Reference](#) section.
7. To update the database version, run the following command:

On Windows operating systems

```
cscript configureDb.vbs --rdbmstype db_type --dbhostname db_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbpassword db_password --dbadminuser db_administrator
--dbadminuserpw db_administrator_password
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype db_type --dbhostname db_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbpassword db_password --dbadminuser db_administrator
--dbadminuserpw db_administrator_password
```

On z/OS operating systems

```
./configureDb.sh --rdbmstype db_type --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
--zlocationname zOS_location_containing_db --zbufferpoolname buffer_pool_in_zOS_location
```

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

8. Start the upgrade by launching the following command:

On Windows operating systems

```
cscript dwcinst.vbs --acceptlicense yes --inst_dir INST_DIR
```

On UNIX operating systems

```
./dwcinst.sh --acceptlicense yes --inst_dir INST_DIR
```

On z/OS operating systems

```
./dwcinst.sh --acceptlicense yes --inst_dir INST_DIR
```

For further details about commands, see [Reference on page 427](#).

9. If you had previously exported the Dynamic Workload Console, as described in [Connecting the Dynamic Workload Console to a new node or database on page 237](#), you can now import them in the new Dynamic Workload Console from the **Administration > Manage Settings** menu. If you have a high availability configuration, import the settings on one node.
10. If you have copied any template `.xml` files from the `templates` folder to the `overrides` folder, check for any differences between the default `.xml` files just upgraded in the `templates` folder and the files you are using in the `overrides` folder. If any differences are present, update the files in the `overrides` folder accordingly.

What to do next

You have now successfully upgraded the Dynamic Workload Console. You can now proceed to [Creating the HCL Workload Automation administrative user on page 266](#).

Creating the HCL Workload Automation administrative user

Instructions to create the HCL Workload Automation administrative user.



HCL Workload Automation administrative user

The HCL Workload Automation administrator creates the administrative user (**wauser**). The administrative user is the user for which the product will be installed in the subsequent steps. This implies that this user has full access to all scheduling objects.

The user name can contain alphanumeric, dash (-), and underscore (_) characters; it cannot contain national characters. The first character of the user name must be a letter.

The following considerations apply:

On Windows operating systems:

- If this user account does not already exist, it is automatically created at installation time.
- If installing on a Windows™ server in a domain, do not define a domain and local ID with the same user name.
- If you specify a domain user, define the name as *domain_name\user_name*.
- If you specify a local user, define the name as *system_name\user_name*. Type and confirm the password.

On UNIX and Linux operating systems:

This user account must be created manually before running the installation and must be enabled to login to the machine where the master domain manager is going to be installed. Create a user with a home directory and group. Use the appropriate UNIX and Linux operating system commands to create the user.



Important: Group names that contain a "/" (forward slash) character can cause permissions to not be set correctly. When HCL Workload Automation retrieves credentials from WebSphere Application Server Liberty, it parses the returned list of groups names assuming they are saved in the format `<realm_name>/<group_name>`. If the group name, the realm name, or both contain a "/" character, the parsing fails.

You can also install HCL Workload Automation using a user different from the root user. This installation method is known as **no-root installation** and applies to all HCL Workload Automation components. Note that if you choose this installation method, only the user who performs the installation can use HCL Workload Automation. For this reason, the typical installation scenario described in this section uses the root user.

For more information, see [HCL Workload Automation user management on page 49](#).

What to do next

You can now proceed to [Upgrading the database for the server components on page 268](#).

Upgrading the database for the server components

Upgrade the database tables before upgrading the server components.



Before you begin



Note: Before upgrading the database schema, ensure you have created a backup. Refer to the documentation related to your RDBMS for information about the backup procedure.



Note: If you are using a PostgreSQL database, check the collation settings before proceeding, as described in [Incorrect collation settings in PostgreSQL database on page 409](#).

Ensure you have acquired information about the HCL Workload Automation tablespaces that were specified when the database tables were created and populated the first time. If values different from the default values were used, then your database administrator must provide them for this upgrade procedure. If default values were used, then they do not need to be specified during the upgrade procedure. The default values for the HCL Workload Automation data, log, and plan tablespaces are as follows:

- **--iwstname** TWS_DATA
 - For Oracle only, the default is `USERS`
- **--iwslogtsname** TWS_LOG
 - For Oracle only, the default is `USERS`
- **--iwsplantsname** TWS_PLAN
 - For Oracle only, the default is `USERS`

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

About this task

You can run the `configureDb` command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

The script creates an SQL file with all the statements needed to upgrade the HCL Workload Automation database schema to the latest version and, by default, automatically applies it.

Default values are stored in the `configureDb<database_vendor>.properties` file, located in `image_location/TWS/interp_name`. For an example of a properties file, see [What is the content of a database properties file? on page 96](#).

If you need to modify any of the default values, edit the `configureDb<database_vendor>.properties` file, but do not modify the `configureDb<database_vendor>.template` file located in the same path.

To upgrade the HCL Workload Automation database schema, perform the following steps:

1. On the workstation where you plan to install the new backup master domain manager or backup dynamic domain manager, extract the HCL Workload Automation package at the latest version to a directory of your choice.
2. Browse to the *image_location/TWS/interp_name* path.
3. Type the following command to upgrade the HCL Workload Automation database schema to the latest version. Ensure that you use the same database administrator credentials you used when the HCL Workload Automation database schema objects were created. The new backup master domain manager or backup dynamic domain manager is configured to point to the existing database instance.

On Windows operating systems

```
cscript configureDb.vbs --rdbmstype db_vendor --dbhostname db_hostname --dbport db_port
--dbname db_name --dbuser db_user --componenttype server_component
--dbadminuser db_administrator --dbadminuserpw db_administrator_password
--iwsname tablespace_data --iwslogtsname tablespace_log --iwsplantsname tablespace_plan
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype db_vendor --dbhostname db_hostname --dbport db_port
--dbname db_name --dbuser db_user --componenttype server_component
--dbadminuser db_administrator --dbadminuserpw db_administrator_password
--iwsname tablespace_data --iwslogtsname tablespace_log --iwsplantsname tablespace_plan
```

where:

--rdbmstype

The database vendor.

--dbhostname db_hostname

The host name or IP address of database server.

--dbport db_port

The port of the database server.

--dbname db_name

The name of the HCL Workload Automation database.

--dbuser db_user

The user that has been granted access to the HCL Workload Automation tables on the database server.

--dbpassword db_password

The password for the user that has been granted access to the HCL Workload Automation tables on the database server. Special characters are not supported.

--dbadminuser db_admin_user

The database administrator user that creates the HCL Workload Automation schema objects on the database server.

--dbadminuserpw *db_admin_password*

The password of the DB administrator user that creates the HCL Workload Automation schema objects on the database server. Special characters are not supported.

--componenttype *MDM | DDM*

The HCL Workload Automation component for which the database is installed. This parameter is optional. Supported values are:

MDM

master domain manager.

DDM

dynamic domain manager.

--iwstsname *tablespace_data*

The name of the tablespace for HCL Workload Automation data. The default value for all supported RDBMS is TWS_DATA, with the exception of Oracle where the default is USERS.

--iwslogtsname *tablespace_log*

The name of the tablespace for the HCL Workload Automation log. The default value for all supported RDBMS is TWS_LOG, with the exception of Oracle where the default is USERS.

--iwsplantsname *db_port*

The name of the tablespace for the HCL Workload Automation plan. The default value for all supported RDBMS is TWS_PLAN, with the exception of Oracle where the default is USERS.

--auth_type *db_name*

The MSSQL authentication mode. The default is SQLSERVER which uses native SQL authentication.

You can optionally point the backup master domain manager to different database residing on the same workstation. For more information, see the topic about Connecting the master domain manager to a new database in *Administration Guide*.



Note: The following parameters specified with the `configureDb` command are also required when you upgrade the server components with the `serverinst` command and their values must be the same:

- **rdbmstype**
- **dbhostname**
- **dbport**
- **dbname**
- **dbuser**
- **dbpassword**

Results

You have now successfully upgraded the database schema for the HCL Workload Automation database.

What to do next

You can now proceed to [Installing a new dynamic domain manager configured as a backup on page 271](#) or to [Installing the new master domain manager configured as a backup on page 276](#).

Installing a new dynamic domain manager configured as a backup

Procedure for installing a dynamic domain manager configured as a backup



Install a new dynamic domain manager at the latest product version level configured as the new backup dynamic domain manager by running the serverinst script.

The procedure to install the dynamic domain manager and backup dynamic domain manager is exactly the same, with the difference that it is performed on two different workstations and that each installation points to its local Open Liberty installation. HCL Workload Automation determines whether or not a dynamic domain manager is already present in the environment and proceeds to install a dynamic domain manager or backup dynamic domain manager accordingly.

The HCL Workload Automation administrator installs the dynamic domain manager as the backup. The following information is required:

Table 14. Required information

Command parameter	Information type	Provided in...
Database information		
--rdbmstype	database type	Upgrading the database for the server components on page 338
--dbhostname	database hostname	
--dbport	database port	
--dbname	database name	
--dbuser	database user name	
--dbpassword	database password	
HCL Workload Automation information		
--wauser	HCL Workload Automation administrative user name	Creating the HCL Workload Automation administrative user on page 337
--wapassword	HCL Workload Automation administrative user password	

Table 14. Required information

(continued)

WebSphere Application Server Liberty Base information		
--wlpdir	WebSphere Application Server Liberty Base installation directory	Installing WebSphere Application Server Liberty on page 315
Security information		
--sslkeyfolder	location of converted certificates	Converting default certificates on page 313
--sslpassword	password of converted certificates	Converting default certificates on page 313

Before starting the installation, ensure the following steps have been completed:

1. [Converting default certificates on page 258](#). Because you are installing a dynamic domain manager, also copy locally the `jwt.crt` file created in the conversion procedure.
2. [Upgrading WebSphere Application Server Liberty on page 260](#) on the workstation where you plan to install the dynamic domain manager and on the workstation where you plan to install the backup dynamic domain manager.
3. [Encrypting passwords \(optional\) on page 262](#)
4. [Upgrading the Dynamic Workload Console and its database on page 264](#)
5. [Creating the HCL Workload Automation administrative user on page 266](#) on the workstations which will host the components at 10.2.5 level.
6. [Upgrading the database for the server components on page 268](#)

You can run the **serverinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. For more information about all **serverinst** parameters and default values, see [Server components installation - serverinst script on page 442](#).

A properties file named `serverinst.properties` is available if you do not want to type parameters in the command line. This is especially useful if you need to specify many parameters or if you want to reuse the file for several installations. The file is located in `image_location/TWS/interp_name`.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

If you need to modify any of the default values, edit the `serverinst.properties` file, but do not modify the `serverinst.template` file located in the same path.

To install the dynamic domain manager as a backup, perform the following steps:

1. Log in to the workstation where you plan to install.
2. Browse to the folder where the `serverinst` command is located:

On Windows operating systems

`image_location\TWS\interp_name`

On UNIX operating systems

`image_location/TWS/interp_name`

3. Start the installation specifying a typical set of parameters. In this case, default values are used for all remaining parameters:

On Windows operating systems

```
cscript serverinst.vbs --acceptlicense yes --rdbmstype db_type
--dbhostname db_hostname --dbport db_port --dbname db_name
--dbuser db_user --dbpassword db_password --wauser wa_user
--wapassword wa_password --componenttype DDM --domain domain_name
--master mdm_name --mdmbrokerhostname mdm_broker_host_name
--sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
--mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir\wlp
```

On UNIX operating systems

```
./serverinst.sh --acceptlicense yes --rdbmstype db_type
--dbhostname db_hostname --dbport db_port --dbname db_name
--dbuser db_user --dbpassword db_password --wauser wa_user
--wapassword wa_password --componenttype DDM --domain domain_name
--master mdm_name --mdmbrokerhostname mdm_broker_host_name
--sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
--mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir/wlp
```

4. Distribute the Symphony file to the new dynamic domain manager configured as backup:
 - a. Ensure that the **optman cf** option is set to *all*.
 - b. To distribute the Symphony file to the new dynamic domain manager configured as backup, run `JnextPlan -for 0000` or wait until the end of the production plan.
 - c. Restore the previous setting of the **optman cf** option, if you previously modified the value.

You have now successfully installed the backup dynamic domain manager at the new product version level.

For more information about all **serverinst** parameters and default values, see [Server components installation - serverinst script on page 442](#).

What to do next

You can now proceed to [Switching the manager to the upgraded backup on page 273](#).

Switching the manager to the upgraded backup

About this task



To switch the back-level manager to the upgraded backup, complete the following procedure:

1. Switch to your upgraded backup manager, which now becomes your current active manager, by issuing the following command from either the Dynamic Workload Console or the **command line** of your back-level manager:

From the Dynamic Workload Console

In the navigation tree, click **Monitoring and Reporting > Monitor Workload >** select the engine and the object type Workstation, click **Run** and, in the table of results, select backup manager workstation name, click **More Actions**, and select **Become Master Domain Manager** or **Become Dynamic Domain Manager**, as necessary.

From the command line of the back-level manager

Issue the following command:

```
conman "switchmgr masterdm;new_mgr_cpu"
```

where *new_mgr_cpu* is the backup manager workstation name.

2. Switch the event processor from the back-level manager to the backup manager, by running the following command from either the Dynamic Workload Console or the **command line** of your back-level manager:

From the Dynamic Workload Console

In the navigation tree, click **Monitoring and Reporting > Monitor Workload >** select the engine and the object type Workstation, click **run** and, in the table of results, select manager workstation name, click **More Actions**, and select **Become Event Processor**.

From the command line of the back-level manager

Issue the following command:

```
conman "switcheventprocessor new_mgr_cpu"
```

where *new_mgr_cpu* is the backup manager workstation name.

Results

You have now successfully switched back-level manager to the upgraded backup.

What to do next

You can now proceed to make this switch permanent, as described in [Making the switch permanent on page 274](#).

Making the switch permanent

Making the switch permanent (DDM)

About this task



In the procedure [Switching the manager to the upgraded backup on page 273](#), you switched your manager promoting your new version backup manager to the role of manager.

To make this configuration fully operational and persistent through **JnextPlan**, complete the following procedure on the new manager, referred to as *new_mgr_cpu*:

1. Edit the *localopts* file and modify the following entry as shown:

```
DEFAULTWS=new_mgr_cpu
```

where *new_mgr_cpu* is the workstation name of the new manager. For more information about *localopts* file, see the section about setting local options in *Administration Guide*.

2. Change the workstation definition of the back-level master by running:

```
composer modify cpu=old_mgr_cpu
```

and in the definition substitute *type=manager* with *type=fta*

3. Change the workstation definition of the new manager by running:

```
composer modify cpu=new_mgr_cpu
```

and in the definition substitute *type=fta* with *type=manager*.

4. Ensure that the **optman** *cf* option is set to *all*.

5. Rebuild the plan to activate the changes to the database:

```
JnextPlan -for 0000
```

6. Restore the previous setting of the **optman** *cf* option, if necessary.

7. Edit the */TWA_DATA_DIR/mozart/globalopts* file and modify the **master=old_mgr_cpu** entry as shown:

```
master=new_mgr_cpu
```

where *new_mgr_cpu* is the workstation name of the new master. For more information about *optman*, see the section about setting global options in *Administration Guide*.

In this way the reports *reptr-pre* and *reptr-post* can run when you run **JnextPlan**.

Results

You have now successfully made the switch permanent.

What to do next

You have now to import to the new dynamic domain manager the security file from the previous dynamic domain manager, as follows:

1. On the previous dynamic domain manager launch the following command to export the security file:

```
dumpsec > <file_name>.txt
```

2. Copy the *<file_name>.txt* file to the new dynamic domain manager.

3. On the new dynamic domain manager, launch the following command to compile and install the security file:

```
makesec <file_name>.txt
```

For more information about the dumpsec and makesec commands, see the topic about updating the security file in *Administration Guide*.

You can now proceed to [Installing the new master domain manager configured as a backup on page 276](#).

Installing the new master domain manager configured as a backup

About this task



You install a master domain manager at the latest product version level configured as the new backup master domain manager by running the serverinst script. The installation process is able to detect the presence of an existing master domain manager and automatically configures this one as the backup master domain manager. The new backup master domain manager is configured to point to the existing database instance.

The HCL Workload Automation administrator installs the master domain manager as the backup. The following information is required:

Table 15. Required information

Command parameter	Information type	Provided in..
Database information		
--rdbmstype	database type	Upgrading the database for the server components on page 338
--dbhostname	database hostname	
--dbport	database port	
--dbname	database name	
--dbuser	database user name	
--dbpassword	database password	
HCL Workload Automation information		
--wauser	HCL Workload Automation administrative user name	Creating the HCL Workload Automation administrative user on page 337

Table 15. Required information

(continued)

--wapassword	HCL Workload Automation administrative user password	
WebSphere Application Server Liberty Base information		
--wlpdir	WebSphere Application Server Liberty Base installation directory	Installing WebSphere Application Server Liberty on page 315
HCL Workload Automation installation directory		
--inst_dir	installation directory	Current procedure
Security information		
--sslkeysfolder	location of converted certificates	Converting default certificates on page 313
--sslpassword	password of converted certificates	Converting default certificates on page 313

You can run the **serverinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. For more information about all **serverinst** parameters and default values, see [Server components installation - serverinst script on page 442](#).

A properties file named `serverinst.properties` is available if you do not want to type parameters in the command line. This is especially useful if you need to specify many parameters or if you want to reuse the file for several installations. The file is located in `image_location/TWS/interp_name`.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

If you need to modify any of the default values, edit the `serverinst.properties` file, but do not modify the `serverinst.template` file located in the same path.

To install the master domain manager as a backup, perform the following steps:

1. Log in to the workstation where you plan to install the master domain manager.
2. Download the installation images from [HCL Software](#).
3. Start the installation specifying a typical set of parameters. In this case, default values are used for all remaining parameters:

On Windows operating systems

```
cscript serverinst.vbs --acceptlicense yes --rdbmstype <db_type>
--dbhostname <db_hostname> --dbport <db_port> --dbname <db_name>
```

```
--dbuser <db_user> --dbpassword <db_password> --wauser <wa_user>
--wapassword <wa_password> --wlpdir <Liberty_installation_dir>\wlp
--sslkeysfolder <certificate_files_path> --sslpassword <keystore_truststore_password>
--inst_dir <installation_dir>
--licenseserverid <license_server_ID>
```

On UNIX operating systems

```
./serverinst.sh --acceptlicense yes --rdbmstype <db_type>
--dbhostname <db_hostname> --dbport <db_port> --dbname <db_name>
--dbuser <db_user> --dbpassword <db_password> --wauser <wa_user>
--wapassword <wa_password> --wlpdir <Liberty_installation_dir>/wlp
--sslkeysfolder <certificate_files_path> --sslpassword <keystore_truststore_password>
--inst_dir <installation_dir>
--licenseserverid <license_server_ID>
```

where

--acceptlicense

Specify **yes** to accept the product license.

--rdbmstype|-r *rdbms_type*

The database type. Supported databases are:

- **DB2**
- **ORACLE** This value applies to Oracle and Amazon RDS for Oracle
- **MSSQL** This value applies to MSSQL and MSSQL cloud-based databases.
- **POSTGRESQL**

This parameter is required and has no default value.

--dbhostname *db_hostname*

The host name or IP address of database server.

--dbport *db_port*

The port of the database server.

--dbname *db_name*

The name of the HCL Workload Automation database.

--dbuser *db_user*

The database user that has been granted access to the HCL Workload Automation tables on the database server.

--dbpassword *db_password*

The password for the user that has been granted access to the HCL Workload Automation tables on the database server. Special characters are not supported.

--wauser *user_name*

The user for which you are installing HCL Workload Automation.

--wapassword *wauser_password*

The password of the user for which you are installing HCL Workload Automation.

On Windows operating systems

Supported characters for the password are alphanumeric, dash (-), underscore (_) characters, and ()|?*~+.@!^

On UNIX operating systems

Supported characters for the password are any alphanumeric, dash (-), underscore (_) characters, and ()|?*~+.

--wlpdir

The path where Open Liberty is installed.

--licenseserverid

The ID of the license server which processes license usage information. This parameter is required. For more information about enabling your product license, see [Enabling product license management on page 52](#).

--sslkeysfolder *keystore_truststore_folder*

The name and path of the folder containing certificates in PEM format. The installation program automatically processes the keystore and truststore files using the password you specify with the **--sslpassword** parameter. The folder must contain the following files:

- **ca.crt**

The Certificate Authority (CA) public certificate. Note that if certificates being installed are part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then this file must contain the Root CA certificate only. Any Intermediate CA certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.



Note: From V10.2.3, if certificates being installed are part of a chain, the `ca.crt` can contain also the intermediate CAs. In this case, it must begin with one or more intermediate CA certificates and end with the Root `ca`.

- **tls.key**

The private key of the end user certificate for the instance to be installed.

- **tls.crt**

The public part of the previous key, that is the end user certificate.

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

You can optionally create a subfolder to contain one or more *.`cert` files to be added to the server truststore as trusted CA, whose name must be `additionalCAs`. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or DB2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**. Note that if the end user certificate being installed in the instance is part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then the Intermediate CAs certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

For further information about how to generate custom certificates, see the topic about managing certificates using Certman in *HCL Workload Automation: Planning and Installation*.

--sslpassword *ssl_password*

The password for the certificates.

For more information, see [sslkeysfolder on page 450](#).

You can optionally encrypt the password using the secure script. For more information, see [Optional password encryption - secure script on page 427](#).

--inst_dir *installation_dir*

The directory of the HCL Workload Automation installation.

--licenseserverid *license_server_ID*

The ID of the license server which processes license usage information. This parameter is required. Instructions about how to obtain the ID of the license server which processes license usage information are provided with the mail confirming your license. For more information, see the section about License computation model in *Administration Guide* and Enabling product license management in *HCL Workload Automation: Planning and Installation*.



Note: The values for the following parameters must match the values you provided when creating and populating the database:

- **--rdbmstype**
- **--dbhostname**
- **--dbport**
- **--dbname**



- **--dbuser**
- **--dbpassword**



Note: Before starting the deployment of a new master domain manager or backup master domain manager on an already used database, be sure that no failed plan creation/extension has been performed. If a failed plan creation or extension has been performed, resolve the failure before attempting the new deployment or unlock the database by running the planman unlock db command.

- If you are installing a backup master domain manager, it is crucial to use the same encryption keys as those on the master domain manager, to ensure it can correctly decrypt encrypted files, such as the Symphony file. To achieve this, perform the following steps:
 - Backup the files located in the `TWA_DATA_DIR\ssl\aes` folder on the backup master domain manager.
 - Copy the files from the `TWA_DATA_DIR\ssl\aes` folder on the master domain manager to the `TWA_DATA_DIR\ssl\aes` folder on the backup master domain manager.

Encryption keys are applicable beginning with version 10.1. If you are upgrading from a version earlier than 10.1, step 4 on page 281 does not apply and can be skipped.

- Run the following command on the back-level master domain manager to add the new backup master domain manager to the plan:

```
JnextPlan -for 0000
```

Results

You have now successfully installed the master domain manager as the backup master domain manager.

What to do next

You can now proceed to [Switching the manager to the upgraded backup on page 281](#).

Switching the manager to the upgraded backup

About this task



To switch the back-level manager to the upgraded backup, complete the following procedure:

- Switch to your upgraded backup manager, which now becomes your current active manager, by issuing the following command from either the Dynamic Workload Console or the **command line** of your back-level manager:

From the Dynamic Workload Console

In the navigation tree, click **Monitoring and Reporting > Monitor Workload >** select the engine and the object type Workstation, click **Run** and, in the table of results, select backup manager workstation name,

click **More Actions**, and select **Become Master Domain Manager** or **Become Dynamic Domain Manager**, as necessary.

From the command line of the back-level manager

Issue the following command:

```
conman "switchmgr masterdm;new_mgr_cpu"
```

where *new_mgr_cpu* is the backup manager workstation name.

2. Switch the event processor from the back-level manager to the backup manager, by running the following command from either the Dynamic Workload Console or the **command line** of your back-level manager:

From the Dynamic Workload Console

In the navigation tree, click **Monitoring and Reporting > Monitor Workload >** select the engine and the object type Workstation, click run and, in the table of results, select manager workstation name, click **More Actions**, and select **Become Event Processor**.

From the command line of the back-level manager

Issue the following command:

```
conman "switcheventprocessor new_mgr_cpu"
```

where *new_mgr_cpu* is the backup manager workstation name.

Results

You have now successfully switched back-level manager to the upgraded backup.

What to do next

You can now proceed to make this switch permanent, as described in [Making the switch permanent on page 282](#).

Making the switch permanent

Making the switch permanent (MDM)

About this task



In the procedure [Switching the manager to the upgraded backup on page 273](#), you switched your manager promoting your new version backup manager to the role of manager.

To make this configuration fully operational and persistent through **JnextPlan**, complete the following procedure on the new manager, referred to as *new_mgr_cpu*:

1. Edit the *localopts* file and modify the following entry as shown:

```
DEFAULTWS=new_mgr_cpu
```

where *new_mgr_cpu* is the workstation name of the new manager. For more information about `localopts` file, see the section about setting local options in *Administration Guide*.

2. Change the workstation definition of the back-level master by running:

```
composer modify cpu=old_mgr_cpu
```

and in the definition substitute *type=manager* with *type=fta*

3. Change the workstation definition of the new manager by running:

```
composer modify cpu=new_mgr_cpu
```

and in the definition substitute *type=fta* with *type=manager*.

4. Ensure that the **optman** `cf` option is set to *all*.
5. Rebuild the plan to activate the changes to the database:

```
JnextPlan -for 0000
```

6. Restore the previous setting of the **optman** `cf` option, if necessary.
7. Edit the `/TWA_DATA_DIR/mozart/globalopts` file and modify the **master=old_mgr_cpu** entry as shown:

```
master=new_mgr_cpu
```

where *new_mgr_cpu* is the workstation name of the new master. For more information about `optman`, see the section about setting global options in *Administration Guide*.

In this way the reports `reptr-pre` and `reptr-post` can run when you run **JnextPlan**.

Results

You have now successfully made the switch permanent.

What to do next

You can now proceed to [Customizing and submitting the optional FINAL job stream on page 283](#).

Customizing and submitting the optional FINAL job stream

About this task



The upgrade process writes the latest FINAL and FINALPOSTREPORTS definitions for the current release in the following file:

`<TWA_HOME>/TWS/config/Sfinal`, where `<TWA_HOME>` is the HCL Workload Automation installation directory. To use these latest definitions, you must merge the functions of your current FINAL and FINALPOSTREPORTS job streams with the syntax of your new FINAL and FINALPOSTREPORTS job streams.



Important: The definitions of the FINAL and FINALPOSTREPORTS job streams in `<TWA_HOME>/TWS/config/Sfinal` are defined on an extended agent that might not be defined in the new environment. If you are planning to use the old definitions to replace the new ones using the `composer replace` command, you must either change the



workstation on which the jobs are defined to an existing one, or you must create a new extended agent where the jobs inside the *Sfinal* are defined.

Complete the following procedure:

1. Depending on your situation, edit your current final job streams and customize the new final job streams as follows:

If you had customized job streams called FINAL and FINALPOSTREPORTS in your database:

- a. Extract the definitions from the current FINAL and FINALPOSTREPORTS job streams file by using composer.
- b. Use a text editor to edit your customized FINAL and FINALPOSTREPORTS job streams.
- c. Merge the job streams with file `<TWA_HOME>/TWS/config/Sfinal` so that the new FINAL and FINALPOSTREPORTS job streams have the same customization as your customized final job streams plus the new required attributes provided by the new FINAL and FINALPOSTREPORTS job streams.
- d. Save your new FINAL and FINALPOSTREPORTS job streams by using composer.

If you had customized final job streams called something other than FINAL and FINALPOSTREPORTS in your database:

- a. Extract the definitions from your customized final job stream files by using composer.
- b. Use a text editor to edit your customized final job stream files.
- c. Merge the job streams with file `<TWA_HOME>/TWS/config/Sfinal` so that the new FINAL and FINALPOSTREPORTS job streams have the same customization as your customized final job streams plus the new required attributes provided by the new FINAL and FINALPOSTREPORTS job streams.
- d. Save these new final job streams so that they have the same names as your current customized final job streams by running the command `composer replace`.

If you had final job streams called something other than FINAL and FINALPOSTREPORTS in your database, but they are not customized:

- a. Make a copy of file `<TWA_HOME>/TWS/config/Sfinal`.
- b. Edit this copy and rename the FINAL and FINALPOSTREPORTS parameters with the actual names.
- c. Run the command `composer replace`.

If you had final job streams called FINAL and FINALPOSTREPORTS in your database, but they are not customized:

Run the command `composer replace <TWA_HOME>/TWS/config/Sfinal`.

If you had final job streams called FINAL and FINALPOSTREPORTS but they are in DRAFT in your database:

Run the command `composer replace` and, after the upgrade, change these job streams into the DRAFT status again.

2. After you customized the new final job streams, you must delete your current final job stream instances (**conman cancel sched** command) and submit the new final job stream instances (**conman sbs sched** command).

During the upgrade, JnextPlan is overwritten even if you customized it. The existing JnextPlan is backed up and renamed to:

On Windows™ operating systems:

JnextPlan.cmd.bk

On UNIX™ and Linux™ operating systems:

JnextPlan.bk

Results

You have now correctly customized and submitted the optional FINAL job stream.

What to do next

You can now proceed to [Installing a new backup dynamic domain manager on page 285](#).

Installing a new backup dynamic domain manager

Procedure for installing a dynamic domain manager configured as a backup.



Install a new dynamic domain manager at the latest product version level configured as the new backup dynamic domain manager by running the `serverinst` script.

The procedure to install the dynamic domain manager and backup dynamic domain manager is exactly the same, with the difference that it is performed on two different workstations and that each installation points to its local Open Liberty installation. HCL Workload Automation determines whether or not a dynamic domain manager is already present in the environment and proceeds to install a dynamic domain manager or backup dynamic domain manager accordingly.

The HCL Workload Automation administrator installs the dynamic domain manager as the backup. The following information is required:

Table 16. Required information

Command parameter	Information type	Provided in...
Database information		

Table 16. Required information

(continued)

--rdbmstype	database type	Upgrading the database for the server components on page 338
--dbhostname	database hostname	
--dbport	database port	
--dbname	database name	
--dbuser	database user name	
--dbpassword	database password	
HCL Workload Automation information		
--wauser	HCL Workload Automation administrative user name	Creating the HCL Workload Automation administrative user on page 337
--wapassword	HCL Workload Automation administrative user password	
WebSphere Application Server Liberty Base information		
--wlpdir	WebSphere Application Server Liberty Base installation directory	Installing WebSphere Application Server Liberty on page 315
Security information		
--sslkeysfolder	location of converted certificates	Converting default certificates on page 313
--sslpassword	password of converted certificates	Converting default certificates on page 313

Before starting the installation, ensure the following steps have been completed:

1. [Converting default certificates on page 258](#). Because you are installing a dynamic domain manager, also copy locally the `jwt.crt` file created in the conversion procedure.
2. [Upgrading WebSphere Application Server Liberty on page 260](#) on the workstation where you plan to install the dynamic domain manager and on the workstation where you plan to install the backup dynamic domain manager.
3. [Encrypting passwords \(optional\) on page 262](#)
4. [Upgrading the Dynamic Workload Console and its database on page 264](#)
5. [Creating the HCL Workload Automation administrative user on page 266](#) on the workstations which will host the components at 10.2.5 level.
6. [Upgrading the database for the server components on page 268](#)
7. [Customizing and submitting the optional FINAL job stream on page 283](#)

You can run the **serverinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. For more information about all **serverinst** parameters and default values, see [Server components installation - serverinst script on page 442](#).

A properties file named `serverinst.properties` is available if you do not want to type parameters in the command line. This is especially useful if you need to specify many parameters or if you want to reuse the file for several installations. The file is located in `image_location/TWS/interp_name`.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

If you need to modify any of the default values, edit the `serverinst.properties` file, but do not modify the `serverinst.template` file located in the same path.

To install the dynamic domain manager as a backup, perform the following steps:

1. Log in to the workstation where you plan to install.
2. Browse to the folder where the `serverinst` command is located:

On Windows operating systems

```
image_location\TWS\interp_name
```

On UNIX operating systems

```
image_location/TWS/interp_name
```

3. Start the installation specifying a typical set of parameters. In this case, default values are used for all remaining parameters:

On Windows operating systems

```
cscript serverinst.vbs --acceptlicense yes --rdbmstype db_type
--dbhostname db_hostname --dbport db_port --dbname db_name
--dbuser db_user --dbpassword db_password --wauser wa_user
--wapassword wa_password --componenttype DDM --domain domain_name
--master mdm_name --mdmbrokerhostname mdm_broker_host_name
--sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
--mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir\wlp
```

On UNIX operating systems

```
./serverinst.sh --acceptlicense yes --rdbmstype db_type
--dbhostname db_hostname --dbport db_port --dbname db_name
--dbuser db_user --dbpassword db_password --wauser wa_user
--wapassword wa_password --componenttype DDM --domain domain_name
--master mdm_name --mdmbrokerhostname mdm_broker_host_name
--sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
--mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir/wlp
```

4. Distribute the Symphony file to the new dynamic domain manager configured as backup:

- a. Ensure that the **optman cf** option is set to *all*.
- b. To distribute the Symphony file to the new dynamic domain manager configured as backup, run `JnextPlan -for 0000` or wait until the end of the production plan.
- c. Restore the previous setting of the **optman cf** option, if you previously modified the value.

You have now successfully installed the backup dynamic domain manager at the new product version level.

For more information about all **serverinst** parameters and default values, see [Server components installation - serverinst script on page 442](#).

What to do next

You can now proceed to [Cleaning up your environment on page 288](#).

Cleaning up your environment

A few final steps towards a clean, efficient environment.

About this task



After performing the steps in the procedure, you might want to clean up the environment by performing the following steps:

1. On the new master at version 10.2.5, modify the new master workstation definitions, both fault-tolerant agent and broker, by setting the **SECUREADDR** and **SECURITYLEVEL** parameters.
2. Enable the full SSL global options, as follows:

```
optman chg sf=yes
```

3. Run `JnextPlan -for 0000` to make the changes effective.
4. When you installed the new master domain manager as a backup of the master at version 9.5, you installed it without SSL enabled, to allow communication with the back-level environment. You have now to enable SSL again, by switching the value of the **eventProcessorEIFPort (ee)** with the value of the **eventProcessorEIFSSLPort (ef)** global options, and vice versa, as follows

```
optman chg ee=<value_of_ef_option>
```

```
optman chg ef=<value_of_ee_option>
```

5. If necessary, move all scheduling objects to the new master domain manager and fault-tolerant agent, as follows

```
composer rename <scheduling_object> <old_master_FTA>#@ <new_master_FTA>#@
```

6. Edit the job definitions to modify the current STREAMLOGON user with the user of the new master domain manager.
7. On the new master domain manager create a backup of the `BrokerWorkstation.properties` file.
8. Copy the `BrokerWorkstation.properties` file from the previous master domain manager and replace the `BrokerWorkstation.properties` file on the new master domain manager, adjusting every key to the values of the new master domain manager.

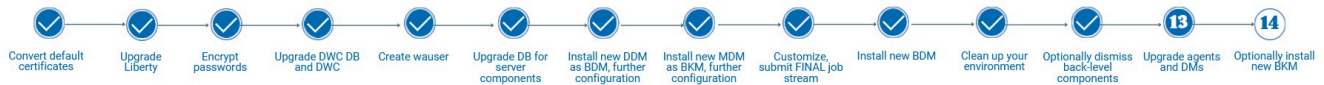
9. Modify the **TCPADDR** and **SECUREADDR** parameters in the broker workstation definition by setting the broker port of the new master domain manager.
10. Modify the **eventProcessorEIFSSLPort** global option to the port of the new master domain manager.
11. Stop and restart WebSphere Application Server Liberty.
12. Run `JnextPlan -for 0000` to make the changes effective.

What to do next

You can now optionally dismiss all back-level components, then proceed to [Upgrading agents and domain managers on page 289](#).

Upgrading agents and domain managers

There are several methods you can choose from to upgrade your agents and domain managers.



The agent upgrade can be performed with minimal impact to scheduling activities. The agents are stopped for the shortest time necessary to perform the maintenance. Any active agent command-line interfaces and processes, such as `conman`, `composer`, `netman`, `mailman`, and `batchman`, to name a few, continue running. Any jobs already running when the upgrade process begins, continue to run as planned, however, no new jobs begin execution during this time. Once the upgrade is complete, the agent is restarted and quickly reconnects with its jobs. Any jobs that were actively running before the upgrade that have not yet completed, continue to run, and any jobs that successfully finished running during the upgrade procedure report a successful job status. An automatic backup and restore feature is in place in case of failure.

If your agents or domain managers are at version 9.4 or 9.5, you can upgrade directly to version 10.2.5.

If you choose to upgrade your environment top-down, then the agents get upgraded progressively after you have upgraded the master domain manager and its backup. This means that new features and enhancements are not available on all of your agents at the same time. If, instead, you choose to upgrade your environment bottom-up, then the agents are upgraded first, and new features and enhancements become available after the master domain manager and its backup have been upgraded.



Important: After upgrading your fault-tolerant agents, it might be necessary to 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. See the topic about updating the security file in the *Administration Guide* for more information.

You can choose to upgrade your agents using any of the following methods:

twinsinst script

A single line command that checks if processes or a command line is running before it starts. It saves disk space and RAM because it is not Java-based. See [Upgrade procedure on page 290](#) and [Upgrading agents on IBM i systems on page 295](#)

Centralized agent update

Upgrade or update multiple fault-tolerant agent and dynamic agent instances at the same time. Download the fix pack installation package, or the elmage upgrade package to the master domain manager and then either run the installation on multiple agent instances or schedule the installation by creating and submitting a job to run. This upgrade method is not supported on z-centric agent instances. See [Centralized agent update on page 300](#).

For a list of supported operating systems and requirements, see the System Requirements Document at [HCL Workload Automation Detailed System Requirements](#).

When the upgrade procedure has completed successfully, the backup instance is deleted.



Note: 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 following path:

On Windows operating systems

`<TWA_home>\TWS`

On UNIX operating systems

`<TWA_DATA_DIR>`

When upgrading dynamic agents featuring both a local and a remote gateway, ensure you either upgrade the agent first and then the gateway or upgrade both at the same time.

After completing the upgrade, you can optionally install a new backup master domain manager at version 10.2.5 to ensure failover capabilities.

Upgrade procedure

Before you begin

1. Verify that the user running the process has the following authorization requirements:

Windows™ operating systems

If you set the Windows User Account Control (UAC), your login account must be a member of the Windows™ **Administrators** group or domain administrators with the right, **Act as Part of the Operating System**.

If you set the Windows User Account Control (UAC) on the workstation, you must run the installation as **administrator**.

UNIX™ and Linux™ operating systems

If the component was installed with root privileges, **root** access is required. If you performed a **no-root installation**, specify the same user used for installing the component.

2. Download the installation images from [HCL Software](#).
3. Ensure that you have enough temporary space before starting the installation process.

About this task

To upgrade agents, from the directory that contains the HCL Workload Automation agent image, run the **twsinst** script using the synopsis described below.

twsinst for Windows™ is a Visual Basic Script (VBS) that you can run in CScript and WScript mode, for example:

```
cscript twsinst.vbs -update -uname user_name -acceptlicense yes -enablefips false
```

On UNIX operating systems, the syntax is as follows:

```
./twsinst -update -uname user_name -acceptlicense yes -enablefips false
```

Synopsis:

Windows™ operating systems

Upgrade an instance

```
./twsinst -update [-uname user_name]
               -acceptlicense yes|no
               [-addjruntime true]
               [-create_link]
               [-inst_dir install_dir [-recovInstReg true]]
               [-lang lang-id]
               [-reset_perm]
               [-patch]
               [-skipbackup]
               [-skipcheckprereq]
               [-skip_usercheck]
               [-wait minutes]
               [-work_dir working_dir]
               [--enablefips true | false]
```

-acceptlicense yes/no

Specify whether or not to accept the License Agreement.

-addjruntime true

Adds the Java™ run time to run job types with advanced options to the agent. The run time environment is used to run application job plug-ins on the agent and to enable the capability to run remotely, from the agent, the dynamic workload broker resource command on the server.

This option is applicable to both fault-tolerant agents and dynamic agents.

By default, if the Java run time was already installed on the agent, it is upgraded.

If the Java run time was not installed on the agent, it is not installed during the upgrade, unless you specify

```
-addjruntime true.
```

If you decided not to install the Java™ run time when you upgrade, you can add this feature later, as described in "Part 2. HCL Workload Automation -> Chapter 7. Configuring -> Adding a feature" in *Planning and Installation Guide*.

-create_link

UNIX™ operating systems only. Create the **symlink** between `/usr/bin/at` and `install_dir/TWS/bin/at`.

For more information, see [Table 2: Symbolic link options on page 34](#).

-enablefips

Specify whether you want to enable FIPS. The default value is `false`. This parameter is optional.

-inst_dir install_dir

The directory where you installed HCL Workload Automation. When upgrading, the directory **inst_dir** is used whether:

- The upgrade process cannot retrieve the product install location from the registries.
- You need to create the HCL Workload Automation registries again before upgrading. See [Re-creating registry files using twsinst on page 370](#) for details.

If you do not provide the **inst_dir** directory and HCL Workload Automation cannot retrieve it from the installation registries, the product is installed in the user home directory.

On Windows™ operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. If not specified, the path is set to %ProgramFiles%\IBM\TWA.

On UNIX™ and Linux™ operating systems:

The path cannot contain blanks. If not specified, the path is set to the *user_name* home directory.

-lang

The language in which the `twinsinst` messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used.



Note: The **-lang** option does not relate to the supported language packs. By default, all supported language packs are installed when you install using the `twinsinst` script.

-password

Windows system only. The password of the user for which you are installing HCL Workload Automation. The password is not required for the upgrade procedure.

-recovInstReg true

To re-create the registry files. Specify if you tried to upgrade a stand-alone, fault-tolerant agent (an agent that is not shared with other components or does not have the connector feature) and you received an error message that states that an instance of HCL Workload Automation cannot be found. This error can be caused by a corrupt registry file. See [Upgrading when there are corrupt registry files on page 370](#). If you specify this parameter you must set **-inst_dir** option.

-reset_perm

UNIX™ systems only. Reset the permissions of the libatrc library.

-skipcheckprereq

If you specify this parameter, HCL Workload Automation does not scan system prerequisites before installing the agent. For more information on the prerequisite check, see [Scanning system prerequisites for HCL Workload Automation on page 48](#).

- patch

Specifies that a patch must be installed. When you specify this option, only the files present in the patch package are replaced in the installed product and all other product files remain unchanged.

-skipbackup

If you specify this parameter the upgrade process does not create a backup of the instance you are upgrading. If the agent upgrade fails, the agent cannot be restored. If you do not specify this parameter, the upgrade process creates a backup of the agent instance in the path `work_dir>/backup`. The `work_dir` is a temporary

directory used by the upgrade process. It can be defined by passing the parameter `-work_dir` to the `twinsinst` script. If you do not define the `work_dir` then by default it is set to `/tmp/TWA_${INST_USER}/tw94`, where `tmp` is the temporary directory of the operating system and `${INST_USER}` is the user performing the upgrade. For example, `/tmp/TWA_jsmith/tw94/backup`.

-skip_usercheck

Enable this option if the authentication process within your organization is not standard, thereby disabling the default authentication option. On UNIX™ and Linux™ operating systems if you specify this parameter, the program skips the check of the user in the `/etc/passwd` file or the check you perform using the `su` command. On Windows™ operating systems if you specify this parameter, the program does not create the user you specified in the `-uname username` parameter. If you specify this parameter you must create the user manually before running the script.

-uname username

The name of the user for which HCL Workload Automation is being updated. The software is updated in this user's home directory. This user name is not to be confused with the user performing the upgrade.

-update

Upgrades an existing agent that was installed using the `twinsinst` script.

-wait minutes

The number of minutes that the product waits for jobs that are running to complete before starting the upgrade. If the jobs do not complete during this interval the upgrade does not proceed and an error message is displayed. Valid values are integers or `-1` for the product to wait indefinitely. The default is **60**.

-work_dir working_dir

The temporary directory used for the HCL Workload Automation upgrade process files deployment.

On Windows™ operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. If you do not manually specify a path, the path is set to `%temp%\TWA\tws<version_number>`, where `%temp%` is the temporary directory of the operating system.

On UNIX™ and Linux™ operating systems:

The path cannot contain blanks. If you do not manually specify a path, the path is set to `/tmp/TWA/tws<version_number>`.

What to do next

When the agent upgrade completes, the agent is restarted and quickly reconnects with its jobs. Any jobs that were actively running before the upgrade that have not yet completed, continue to run, and any jobs that successfully finished running during the upgrade procedure report a successful job status. An automatic backup and restore feature is in place in case of failure.

Examples

About this task

This section contains examples of **twsinst** scripts that you can use to upgrade an agent.

To upgrade an agent installed in the user home directory that does not have the dynamic scheduling capabilities and the Java™ run time to run job types with advanced options:

```
./twsinst -update -uname twsuser -acceptlicense yes
```

To upgrade an agent installed in the path `/opt/IBM/TWA` on UNIX operating systems and in the path `C:\Program Files\IBM\TWA` on Windows operating systems, and give it dynamic scheduling capabilities, but not the Java™ run time to run job types with advanced options:

On Windows™ operating systems:

```
cscript twsinst -update -uname TWS_user -password password
-acceptlicense yes
-tdwbhostname mybroker.mycompany.com -tdwbport 31116
-inst_dir "c:\Program Files\IBM\TWA"
```

On UNIX™ and Linux™ operating systems:

```
./twsinst -update -uname twsuser
-acceptlicense yes
-tdwbhostname mybroker.mycompany.com
-tdwbport 31116 -inst_dir /opt/IBM/TWA
```

To upgrade an agent and give it both dynamic scheduling capabilities and the Java™ run time to run job types with advanced options. The run time environment is used to run application job plug-ins on the agent and to enable the capability to remotely run, from the agent, the dynamic workload broker resource command on the server:

On Windows™ operating systems:

```
cscript twsinst -update -uname TWS_user -password password
-acceptlicense yes
-tdwbhostname mybroker.mycompany.com -tdwbport 31116 -addjruntime true
-inst_dir "c:\Program Files\IBM\TWA"
```

On UNIX™ and Linux™ operating systems:

```
./twsinst -update -uname twsuser -acceptlicense yes
-tdwbhostname mybroker.mycompany.com
-tdwbport 31116 -addjruntime true
```

Upgrading agents on IBM i systems

How to upgrade agents on IBM i systems.

About this task

You can upgrade the agent on an IBM i system by using the `twsinst` installation script.

If you plan to enable FIPS, ensure your certificates meet FIPS standards before getting started.

To ensure FIPS compliance, all HCL Workload Automation components must be at version 10.2.5 or later, certificates must employ at least a robust 2K RSA key and use encryption algorithms different from `MD5-RSA` and `SHA1-RSA`.

To upgrade an HCL Workload Automation agent, perform the following steps:

1. Sign on as the user who performed the installation, either **QSECOFR** or an existing user with ALLOBJ authority. If you installed with a user different from **QSECOFR**, use the same user who performed the installation and specify the **allObjAuth** parameter to indicate that the user has the ALLOBJ authority. For more information about this parameter, see [Agent installation parameters on IBM i systems on page 159](#). You can find the name of the profile used to perform the installation in the `instUser` located in the `agent_data_dir/installation/instInfo`.
2. Download the installation images from [HCL Software](#).
3. If you downloaded the elimages, to extract the package, use the *PASE* shell or the *AIXterm* command.

Using *PASE* shell:

- a. Open the *PASE* shell.
- b. Run the command "CALL QP2TERM".
- c. Locate the folder where you downloaded the elimages and run the command:

```
"tar xvf TWS1025_IBM_I.tar"
```

- d. Exit from the *PASE* shell.

Using *AIXterm* command:

- a. Start the *Xserver* on your desktop.
- b. On the iSeries machine, open a *QSH shell* and export the display.
- c. In *QSH shell*, go to the directory `/QopenSys` and run the command "aixterm -sb".
- d. A pop-up window is displayed on your desktop. By Using this pop-up window, extract the file `TWS1025_IBM_I.tar`.

4. Open a *QSH shell* and run the **twsinst** script.

The installation procedure replaces the library to the user profile library list of the dynamic agent user profile and sets this job description as the job description of the dynamic agent user profile. The upgrade process replaces the new version of the agent in the directory where the old agent is installed.

If the operation fails to understand the cause of the error, see [Analyzing return codes for agent installation, upgrade, restore, and uninstallation on page 400](#).

Command usage and version

Show command usage and version

```
twsinst -u | -v
```

Upgrade an instance

```
./twsinst -update -uname user_name
-acceptlicense yes|no
[-addjruntime true]
[-allObjAuth]
```

```

[-create_link]
[-hostname host_name]
[-inst_dir install_dir]
[-jimport port_number]
[-jimportssl boolean]
[-lang lang-id]
[-reset_perm]
[-recovInstReg true]
[-skip_usercheck]
[-tdwbhostname host_name]
[-tdwbport port_number]
[-wait minutes]
[-work_dir working_dir]

```

For a description of the installation parameters and options that are related to agent on this operating system, see [Agent upgrade parameters on IBM i systems on page 297](#).

Agent upgrade parameters on IBM i systems

About this task

The parameters set when using the **twinst** script to upgrade a dynamic agent on IBM i systems.

-acceptlicense yes/no

Specifies whether to accept the License Agreement.

-addjruntime true

Adds the Java™ run time to run job types with advanced options to the agent. The run time environment is used to run application job plug-ins on the agent and to enable the capability to run remotely, from the agent, the dynamic workload broker resource command on the server.

By default, if the Java run time was already installed on the agent, it will be upgraded to the new version.

If the Java run time was not installed on the agent, it will not be installed during the upgrade, unless you specify `-addjruntime true`.

If you decided not to install Java™ run time when you upgrade, you can still add this feature later. For details about how to add a feature, see *HCL Workload Automation for Z: Planning and installation*.

-allObjAuth

If you are installing, upgrading, or uninstalling with a user different from the default **QSECOFR** user, this parameter specifies that the user has the required ALLOBJ authority. Ensure the user is existing and has ALLOBJ authority because the product does not verify that the correct authority is assigned. The same user must be specified when installing, upgrading or uninstalling the agent. If you are using the **QSECOFR** user, this parameter does not apply.

-create_link

Create the **symlink** between `/usr/bin/at` and `<install_dir>/TWS/bin/at`. See [Table 2: Symbolic link options on page 34](#) for more information.

-displayname

The name to assign to the agent. The default is the host name of this computer.

-inst_dir *installation_dir*

The directory of the HCL Workload Automation installation.



Note: The path cannot contain blanks. If you do not manually specify a path, the path is set to the default home directory, that is, the *user_home\user_name* directory.

-jimport *port_number*

The JobManager port number used by the dynamic workload broker to connect to the HCL Workload Automation dynamic agent. The default value is **31114**. The valid range is from 1 to 65535.

-jimportssl *true/false*

The JobManager port used by the dynamic workload broker to connect to the HCL Workload Automation dynamic agent. This number is registered in the *ita.ini* file located in the *ITA/cpa/ita* directory.

For communication using SSL or HTTPS

Set **jimportssl = true**. To communicate with the dynamic workload broker, it is recommended that you set the value to **true**. If the value is set to *true*, the port specified in **jimport** communicates in HTTPS.

For communication without using SSL, or through HTTP

Set **jimportssl = false**. If the value is set to *false*, the port specified in **jimport** communicates in HTTP.

-lang *lang_id*

The language in which the *twinst* messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used.



Note: This is the language in which the installation log is recorded, and not the language of the installed engine instance. The *twinst* script installs all languages by default.

-recovInstReg *true*

To re-create the registry files. Specify it if you have tried to upgrade a stand-alone agent and you received an error message that states that an instance of HCL Workload Automation cannot be found, this can be caused by a corrupt registry file. See [Upgrading when there are corrupt registry files on page 370](#).

-skip_usercheck

Enable this option if the authentication process within your organization is not standard, thereby disabling the default authentication option. If you specify this parameter, you must create the user manually before running the script.

-skipcheckprereq

If you specify this parameter, HCL Workload Automation does not scan system prerequisites before upgrading the agent.

For a detailed list of supported operating systems and product prerequisites, see [HCL Workload Automation Detailed System Requirements](#).

-tdwbhostname *host_name*

The dynamic workload broker fully qualified host name. It is used together with the **-tdwbport** *tdwbport_number* parameter. It adds and starts the capabilities to run workload dynamically to HCL Workload Automation. This value is registered in the **ResourceAdvisorUrl** property in the `JobManager.ini` file.

-tdwbport *tdwbport_number*

The dynamic workload broker HTTP or HTTPS port number used to add dynamic scheduling capabilities to your distributed or end-to-end environment. It is used together with the **-tdwbhostname** *host_name* parameter. This number is registered in the **ResourceAdvisorUrl** property in the `JobManager.ini` file. Specify a nonzero value to add dynamic capability. The valid range is 0 to 65535.

-uname *user_name*

The name of the user for which HCL Workload Automation is being updated. The software is updated in this user's home directory. This user name is not to be confused with the user performing the upgrade.



Note: This user name is not the same as the user performing the installation logged on as **QSECOFR**.

-update

Upgrades an existing agent that was installed using **twsinst**.

-wait *minutes*

The number of minutes that the product waits for jobs that are running to complete before starting the upgrade. If the jobs do not complete during this interval the upgrade does not proceed and an error message is displayed. Valid values are integers or **-1** for the product to wait indefinitely. The default is **60** minutes.

-work_dir *working_dir*

The temporary directory used for the HCL Workload Automation installation process files deployment. The path cannot contain blanks. If you do not manually specify a path, the path is set to `/tmp/TWA/tws1025`.

Example upgrade of an agent on IBM i systems

About this task

The following example shows the syntax used when using the **twsinst** script to upgrade an instance of the agent on IBM i system.

```
./twsinst -update
-uname TWS_user
-allObjAuth
```

```
-acceptlicense yes
-nobackup
-work_dir "/tmp/TWA/tws1025"
```

The twsinst script log files on IBM i systems

About this task

The twsinst log file name is:

Where: `<TWS_INST_DIR>/twsinst_IBM_i_TWS_user^product_version.log`

TWS_INST_DIR

The HCL Workload Automation installation directory. The default installation directory is `/home/TWS_user`.

TWS_user

The name of the user for which HCL Workload Automation was installed, that you supplied during the installation process.

product_version

Represents the product version. For example, for version 10.2.5 of the product, the value is 10.2.5.00

Centralized agent update

You can install fix packs or upgrade releases for multiple fault-tolerant agent and dynamic agent instances, by downloading a package on the master domain manager workstation and updating the multiple agent instances by running an action from the Dynamic Workload Console.

You can also schedule the centralized update of multiple agent instances, by using the Dynamic Workload Console or the command line.

The centralized agent update process does not apply to z-centric agents. Also, a distributed master domain manager is required.

During the upgrade or update, the agents are stopped for the shortest time necessary to perform the maintenance. Any active agent command-line interfaces and processes, such as conman, composer, netman, mailman, and batchman, to name a few, continue running. Any jobs already running when the upgrade process begins, continue to run as planned, however, no new jobs begin execution during this time. Once the upgrade is complete, the agent is restarted and quickly reconnects with its jobs. Any jobs that were actively running before the upgrade that have not yet completed, continue to run, and any jobs that successfully finished running during the upgrade procedure report a successful job status.

You can find the full procedure in [Centralized agent update by using Dynamic Workload Console on page 301](#).



Note:

Avoid installing multiple agents (fault-tolerant agents or dynamic agents) at the same time on the same system, because this could cause the installation to fail.



For the latest information about Centralized agent update, see the Release Notes available at [HCL Workload Automation Release Notes](#).

Centralized agent update by using Dynamic Workload Console

You can centrally update multiple fault-tolerant agent and dynamic agent instances with just one single action by using Dynamic Workload Console.

Before you begin

For more information about the `manage` keyword usage, see the section about object type - `cpu` in *Administration Guide*. For an example of a master domain manager `Security` file, see the section about the security file on the master domain manager to install fix packs or upgrade fault-tolerant agents and dynamic agents in *Administration Guide*.

About this task

Complete the following steps:

1. From the installation package download site, download on the master domain manager workstation the fix pack or upgrade installation package that you want to install on fault-tolerant agent or dynamic agent instances in the following default directory:

On Windows operating systems:

```
<TWA_home>\TWS\depot\agent
```

On UNIX operating systems:

```
<TWA_home>/TWS/depot/agent
```

where `TWA_home` is the master domain manager installation directory.

You can change the default directory value performing the following steps:

- Stop WebSphere Application Server Liberty Base on the master domain manager
- Modify the `com.ibm.tws.conn.engine.depot` key value in the following property file:

On Windows operating systems:

```
TWA_home>\usr\servers\engineServer\resources\properties  
\TWSConfig.properties
```

On UNIX operating systems:

```
TWA_home>/usr/servers/engineServer/resources/properties/  
TWSConfig.properties
```

- Start WebSphere Application Server Liberty Base

Ensure the installation files are readable by the operating system user which owns the Application Server process (java).

2. Log on to Dynamic Workload Console.

3. Create a **Monitor Workstations** task, as described in the section about creating a task to Monitor Workstations in *Dynamic Workload Console User's Guide*.
4. Run a **Monitor Workstations** task and select one or more dynamic agent or fault-tolerant agent instances that you want to update.
5. Click **More Actions > Update agent**. The Update agent action checks whether the selected agent is a supported workstation type.

The Update agent action is applicable to the following workstation types only:

- Dynamic Agent
- Fault-tolerant agent

The Update agent action is not applicable to the following workstation types:

- Master domain manager
- Backup master domain manager
- Dynamic domain manager
- Backup dynamic domain manager
- Extended agent
- Standard agent
- Remote engine
- Broker
- Pool
- Dynamic pool
- Limited fault-tolerant agent

The process updates the agent only if the workstation type is supported. Otherwise, either an error message is displayed on the Dynamic Workload Console, or is written in the operator log messages console, depending on the workstation type.

You can schedule the centralized update of multiple agent instances, by using the Dynamic Workload Console or the command line. For a description of the scheduling option, see: [Scheduling the centralized agent update on page 303](#).

For a description of the **Update agent** action on fault-tolerant agents and dynamic agents, see: [Updating fault-tolerant agent and dynamic agent instances on page 305](#).

Results

Verify the update agent results by completing one of the following actions in the Dynamic Workload Console:

Check the operator log messages console:

Click **Monitoring and Reporting > Event Monitoring > Monitor Triggered Actions** and check the messages related to the agent workstation update.

The following event rules are triggered:

UPDATESUCCESS

When the workstation is successfully updated

UPDATEFAILURE

When an error occurs

UPDATERUNNING

With the information about the update process status

Check the workstation version changes:

After the next plan update, in the `Monitor Workstations` view of the Dynamic Workload Console, you can check the updated version in the `Version` column of the selected agent. Otherwise, if you do not want to wait for the next plan update to see the updated version, run the command **JnextPlan -for 0000** with the **-noremove** option.

You can also perform a **manual check of the update agent results** by looking at the following log files on the agent system:

On Windows operating systems:

```
<TWA_home>\logs\centralized_update.log
```

On UNIX operating systems:

```
<TWA_home>/logs/centralized_update.log
```

Scheduling the centralized agent update

About this task

You can schedule the centralized update of multiple agent instances by creating a centralized agent update job, either by using the Dynamic Workload Console or the **composer** command line.

Creating a centralized agent update job by using the Dynamic Workload Console:

1. Log on to the Dynamic Workload Console.
2. Create a `Centralized agent update` job type definition, as described in "Creating job definitions" in *Dynamic Workload Console User's Guide*.
3. In the properties panel, specify the attributes for the job definition that you are creating. For all the details about available fields and options, see the online help by clicking the "?" in the upper-right corner.
4. In the `Connection` tab, specify the master domain manager workstation where you loaded the fix pack installation package, or the upgrade elmage, that you want to install on fault-tolerant agent or dynamic agent instances.
5. In the `Action` tab, define the list of fault-tolerant agent or dynamic agent instances that you want to update. You can select up to 20 agent instances.
6. Save the job definition in the database.

Creating a centralized agent update job by using the composer command line:

This section describes the required and optional attributes that you need to specify to create a centralized agent update job by using the **composer** command line. For more information, see "Job definition" in *User's Guide and Reference*:

Table 17. Required and optional attributes for the definition of a centralized agent update job

Attribute	Description and value	Required
hostname	The host name of the master domain manager workstation where you loaded the fix pack installation package, or the upgrade image, that you want to install on fault-tolerant agent or dynamic agent instances.	✓
port	The port number of the master domain manager workstation.	✓
protocol	The protocol for connecting to the master domain manager workstation. Supported values are http and https .	✓
userName	The user to be used for accessing the master domain manager workstation. This attribute is optional, depending on your settings.	
password	The password to be used for accessing the master domain manager workstation. This attribute is optional, depending on the settings on your server.	
NumberOfRetries	The number of times the program tries to connect to the master domain manager workstation. Default value is 0.	
RetryIntervalSeconds	The number of seconds the program waits before retrying the operation. Default value is 30 seconds.	
workstationListValues	<p>The list of agent instances that you want to update.</p> <p>Example:</p> <pre><jsdlcentralizedagentupdate:workstationsListValue> NY053015_AGT (type: Agent, version: 9.5.0.00) </jsdlcentralizedagentupdate:workstationsListValue> <jsdlcentralizedagentupdate:workstationsListValue> NY053009_AGT (type: Agent, version: 9.5.0.00)< /jsdlcentralizedagentupdate:workstationsListValue> <jsdlcentralizedagentupdate:workstationsListValue> NY053016_FTA (type: FTA, version: 9.5.0.00) </jsdlcentralizedagentupdate:workstationsListValue></pre> <p>You can specify up to 20 agent instances.</p>	✓

Scheduling a centralized agent update job

You can schedule a centralized agent update job by adding the necessary scheduling arguments to your job, and submitting it. You can submit jobs by using the Dynamic Workload Console or the **conman** command line.

When the job runs, the job forwards to the master domain manager the Update agent request for all the fault-tolerant agent or dynamic agent instances that you selected, and then completes.



Note: The job does not wait for the Update agent request to complete. The completion status of the centralized agent update job refers only to the submission of the Update agent request; the completion status does not refer to the agent update results. To verify the agent update results, see the *Results* section in [Centralized agent update by using Dynamic Workload Console on page 301](#).

Job properties

When the job completes, you can see the job properties by running:

```
conman sj job_name:jobprop
```

where *job_name* is the centralized agent update job name.

The following example shows the `Extra Information` section of the output command:

```
EXTRA INFORMATION
The update request has been successfully submitted for the following workstations:
NY053015_AGT|NY053009_AGT|NY053016_FTA
```

Updating fault-tolerant agent and dynamic agent instances

A description of the **Update agent** action on fault-tolerant agents and dynamic agents.

About this task

When you run the `update agent` action in the `Monitor Workstations` task from Dynamic Workload Console, or when you schedule a centralized agent update job, HCL Workload Automation completes the following steps:

1. The fix pack or upgrade installation package is copied to the master domain manager workstation, and its content is extracted to the following default directory:

For fault-tolerant agent workstations:

On Windows™ operating systems:

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

On UNIX™ operating systems:

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

For dynamic agent workstations:

On Windows™ operating systems:

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

On UNIX™ operating systems:

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

Where *TWA_home* is the fault-tolerant agent or dynamic agent installation directory. You can change this default directory by modifying the `DownloadDir` value in the following configuration file:

For fault-tolerant agent workstations:

On Windows™ operating systems:

```
<TWA_home>\localopts
```

On UNIX™ operating systems:

```
<TWA_DATA_DIR>/TWS/localopts
```

For dynamic agent workstations:

On Windows™ operating systems:

```
<TWA_home>\TWS\ITA\cpa\config\JobManager.ini
```

On UNIX™ operating systems:

```
<TWA_DATA_DIR>/ITA/cpa/config/JobManager.ini
```

**Note:**

If the path specified in `DownloadDir` does not exist, a warning message is issued and the default download directory is used.

If you are updating both fault-tolerant agent and dynamic agent instances on the same workstation, be sure that you specify different download directories.

2. On the agent workstation, the following script runs automatically:

For fault-tolerant agent workstations:**On Windows™ operating systems:**

```
<TWA_home>\TWS\stdlist\download\.self\selfupdate.wsf
```

On UNIX™ operating systems:

```
<TWA_DATA_DIR>/stdlist/download/.self/selfupdate.sh
```

For dynamic agent workstations:**On Windows™ operating systems:**

```
<TWA_home>\TWS\stdlist\JM\download\.self\selfupdate.wsf
```

On UNIX™ operating systems:

```
<TWA_DATA_DIR>/stdlist/JM/download/.self/selfupdate.sh
```

The centralized agent update script, named **selfupdate**, performs a backup of the agent workstation, runs the **twsinst** installation command, and creates the following log file:

On Windows™ operating systems:

```
<TWA_home>\TWS\logs\centralized_update.log
```

On UNIX™ operating systems:

```
<TWA_DATA_DIR>/TWS/logs/centralized_update.log
```

**Note:**

If for any reason the agent update fails, the **selfupdate** script restores the agent to its initial status. The backup files are removed after the agent update completes successfully. The backup files are not removed when the agent restore fails or is successful. For more information about restoring agent instances, see the



troubleshooting scenario [Manually restore agent instances when the automatic restore fails on page 308](#).

To modify the backup directory, specify the new directory in the BACKUP_DIR variable in the selfupdate.wsf script.

Troubleshooting scenarios

You can troubleshoot the centralized agent update.

You can troubleshoot the centralized agent update by reading the centralized_update log file.

Prerequisite scan detects missing prerequisites and the centralized agent update fails

You are centrally updating dynamic agents or fault-tolerant agents but the prerequisite scan detects missing prerequisites and the agent installation fails.

Cause and solution

The centralized agent update fails because the prerequisite scan detects missing prerequisites. In this case, analyze the prerequisite scan log file and solve the error, if any. You can then decide to rerun the installation or upgrade without executing the prerequisite scan. To do this, perform the following steps:

1. On the master domain manager workstation, go to the directory where you download the fix pack installation package, or the image that you want to install on the agent. The default directory value is:

On Windows operating systems:

```
<TWA_home>\TWS\depot\agent
```

On UNIX operating systems:

```
<TWA_home>/TWS/depot/agent
```

where *TWA_home* is the master domain manager installation directory.

2. Edit the following script:

On Windows operating systems:

```
<TWA_home>\TWS\depot\agent\TWS1025_agent_platform_AGENT.zip\self
\selfupdate.wsf
```

On UNIX operating systems:

```
<TWA_home>/TWS/depot/agent/TWS1025_agent_platform_AGENT.zip/.self/
selfupdate.sh
```

3. In the selfupdate script, locate the twsinst installation command and add the `-skipcheckprereq` option. If you specify the `-skipcheckprereq` parameter, the twsinst script does not execute the prerequisite scan. For more information about the `-skipcheckprereq` option, see [Agent installation parameters - twsinst script on page 119](#).

Centralized agent update fails because the temporary backup directory is too small

You are centrally updating dynamic agents or fault-tolerant agents but the backup directory used is too small, and the agent installation fails.

Cause and solution

The centralized agent update fails because the backup directory, by default */tmp*, does not have enough space. You can set a different directory by performing the following steps:

1. On the master domain manager workstation, go to the directory where you downloaded the fix pack installation package, or the image that you want to install on the agent. The default directory value is:

On Windows operating systems:

```
<TWA_home>\TWS\depot\agent
```

On UNIX operating systems:

```
<TWA_home>/TWS/depot/agent
```

where *TWA_home* is the master domain manager installation directory.

2. Edit the following script:

On Windows operating systems:

```
<TWA_home>\TWS\depot\agent\TWS1025_agent_platform_AGENT.zip\self  
\selfupdate.wsf
```

On UNIX operating systems:

```
<TWA_home>/TWS/depot/agent/TWS1025_agent_platform_AGENT.zip/self/  
selfupdate.sh
```

3. In the selfupdate script, locate the BACKUP_DIR variable and replace the value to the directory you want to use as backup.



Note: This directory will be removed at the end of the installation.

Manually restore agent instances when the automatic restore fails

You are upgrading dynamic agents or fault-tolerant agents using either the centralized agent update method or the *twinst* script, but the update process fails and starts the automatic restore process. If the automatic restore process fails, you need to manually restore the old agent instances.

Cause and solution

The automatic restore process might fail for several causes, for example, the automatic process does not have the necessary space to perform the operation. If you want to manually restore the old agent instance, complete the following steps:

1. On the workstation where the agent is installed, go to the temporary directory, where the selfupdate script backs up the agent installation directory. The default temporary directory value is:

Centralized agent update method

On Windows operating systems:

```
%TEMP%\backupTWS\date
```

On UNIX operating systems:

```
/tmp/backupTWS/date
```

Where *date* is the date of the selfupdate running for your agent instance.

twinsinst script upgrade method

On Windows operating systems:

```
<working_dir>\backupTWS\date
```

On UNIX operating systems:

```
<working_dir>/backupTWS/date
```

where *working_dir* is a temporary directory used by the upgrade process. You define the *working_dir* by passing the **-work_dir** parameter to the twinsinst script. If you do not define the *working_dir* then by default it is set to `/tmp/TWA_${INST_USER}/tws94`, where *tmp* is the temporary directory of the operating system and `${INST_USER}` is the user performing the upgrade. For example, on a UNIX operating system: `/tmp/TWA_jsmith/tws94/backup`.

Where *date* is the date of the selfupdate running for your agent instance.

2. Locate the *agent_instance_backup_dir* backup directory for your agent instance.
3. Copy the content of the following directory to the *TWS_agent_inst_dir* agent installation directory:

Centralized agent update method

On Windows operating systems:

```
%TEMP%\backupTWS\date\agent_instance_backup_dir
```

On UNIX operating systems:

```
/tmp/backupTWS/date/agent_instance_backup_dir
```

twinsinst script upgrade method

On Windows operating systems:

```
<working_dir>\backupTWS\date\agent_instance_backup_dir
```

On UNIX operating systems:

```
<working_dir>/backupTWS/date/agent_instance_backup_dir
```

4. In the *TWS_agent_inst_dir* directory, re-create the *stdlist* directory.
5. Manually delete the following lock file:

Centralized agent update method

On Windows operating systems:

```
%TEMP%\twsselfupdate.lock
```

On UNIX operating systems:

```
/tmp/twsselfupdate.lock
```

twsinst script upgrade method

On Windows operating systems:

```
<working_dir>\twsselfupdate.lock
```

On UNIX operating systems:

```
<working_dir>/twsselfupdate.lock
```

6. Restart the agent instance.

Centralized agent update does not complete and no operator message is displayed

You are centrally updating dynamic agents and fault-tolerant agents from Dynamic Workload Console. An agent is in running status in the Dynamic Workload Console, but the update process does not complete and no operator message is displayed.

Cause and solution

The agent has been stopped but the Dynamic Workload Console has not been refreshed yet and reports an incorrect agent status. When the update agent action is selected on this agent, the process cannot start and no operator message is displayed.

To solve this problem, you have to check the agent status locally and restart the agent instance if needed. Then, you have to re-issue the update agent command.

No Monitor Operator Messages available when updating with Centralized agent update process in SSL mode

You are centrally updating dynamic agents and fault-tolerant agents from Dynamic Workload Console. An agent is in running status in the Dynamic Workload Console, but the update process does not complete and no operator message is displayed.

Cause and solution

When running the Centralized Agent Update from Version 95 Fix Pack 5 to the current version with the event processor configured in SSL mode, no Monitor Operator Messages are displayed for either fault-tolerant agents and dynamic agents.

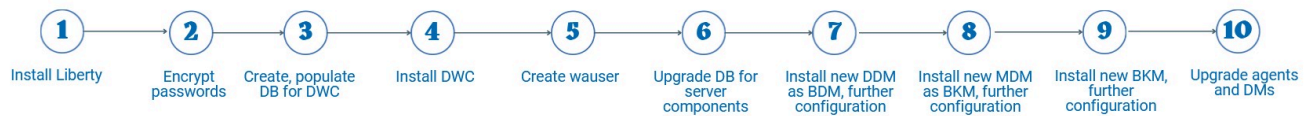
To solve this problem, perform one of the following steps:

- Perform centralized update from Version 95 Fix Pack 5 to Version 10 in SSL mode.
- Disable SSL communication for the event processor before running Centralized agent update, as follows:

1. Stop WebSphere Application Server Liberty Base.
2. Use the **eventProcessorEIFPort** optman option to define the port to be used for event processor communication. This automatically disables communication in SSL mode.
3. Start WebSphere Application Server Liberty Base.
4. Run JnextPlan to make the change effective.
5. Perform Centralized agent update from Version 95 Fix Pack 5 to the current version.

Parallel upgrade from version 9.4.0.x to version 10.2.5

Detailed steps to perform a parallel upgrade from version 9.4.0.x to version 10.2.5



A number of major product improvements have been inserted starting from version 9.5. For this reason, when upgrading from version 9.4.0.x, you have to perform a fresh installation of the following components and prerequisites:

- WebSphere Application Server Liberty Base
- Dynamic Workload Console
- Dynamic Workload Console database
- master domain manager
- backup master domain manager
- dynamic domain manager
- backup dynamic domain manager

Before you start the upgrade, ensure you have performed the following procedures:

- [Connecting the Dynamic Workload Console to a new node or database on page 237](#). If you are currently using Derby, install a supported database before exporting Dynamic Workload Console data. This is necessary because Derby is no longer supported as of version 10.2.3.
- Installing the fix for APAR IJ47731 on the master domain manager. To obtain the fix for your product version, contact Software Support.
- [Configuring TLS to the appropriate version on page 312](#) on the 9.4 master domain manager to ensure communication in your environment.
- [Converting default certificates on page 313](#), if you are using default certificates in your current environment. Use this procedure to convert the certificates from the JKS to the PEM format, then copy them to the workstations where you plan to install the server components (dynamic domain manager and its backups, master domain manager and its backups) and the Dynamic Workload Console.

Certificates are now required when installing or upgrading HCL Workload Automation. You can no longer install nor upgrade HCL Workload Automation without securing your environment with certificates. The required certificates are:

- ca.crt
- tls.key
- tls.crt

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

- Install the latest fix pack for version 9.4 on all workstations in your environment.

To perform a parallel upgrade from version 9.4.0.x to 10.2.5, perform the following steps:

1. [Installing WebSphere Application Server Liberty on page 315](#) on the workstations hosting the Dynamic Workload Console and the server components. This is a prerequisite component which replaces WebSphere Application Server used in version 9.4.0x.
2. [Encrypting passwords \(optional\) on page 317](#)
3. [Creating and populating the database for the Dynamic Workload Console on page 319](#)
4. [Installing the Dynamic Workload Console on page 334](#)
5. [Creating the HCL Workload Automation administrative user on page 337](#) on the workstations which will host the components at 10.2.5 level.
6. [Upgrading the database for the server components on page 338](#)
7. [Installing a new dynamic domain manager configured as a backup on page 341](#)
 - a. [Installing a new dynamic domain manager configured as a backup on page 341](#)
 - b. [Switching the dynamic domain manager to the new dynamic domain manager configured as backup on page 344](#)
 - c. [Installing a new backup dynamic domain manager on page 345](#) to replace the backup dynamic domain manager which you have switched to become the current dynamic domain manager.
 - d. [Switching back to the old dynamic domain manager \(optional\) on page 348](#)
8. [Installing the new master domain manager configured as a backup on page 348](#)
 - a. [Ensuring communication in your environment on page 364](#)
 - b. [Switching the master domain manager to the new backup master on page 355](#)
 - c. [Making the switch manager permanent on page 356](#)
 - d. [Customizing and submitting the optional FINAL job stream on page 358](#)
9. [Installing a new backup master domain manager on page 359](#)
 - a. [Installing a new backup master domain manager on page 360](#)
 - b. [Uninstalling the back-level backup master domain manager on page 365](#)
10. [Upgrading agents and domain managers on page 366](#)

Configuring TLS to the appropriate version

Transport Layer Security (TLS) is a cryptographic protocol designed to provide secure communication over a computer network. It ensures that data transmitted between applications, such as web browsers and servers, remains private and tamper-proof. Setting TLS to version 1.2 is required to ensure communication between 9.4 and 10.2.5 components.

In back-level environments, for example 9.4, SSL is not enabled by default and TLS version 1.2 needs to be enabled on the back-level master domain manager to enable communication. Perform the following steps on the back-level master domain manager:

1. Browse to the `<JazzSMHome>/profile/config/cells/JazzSMNode01Cell` path, where

<JazzSMHome>

is the directory where Jazz for Service Management is installed.

2. Open the `security.xml` file in a flat-text editor.
3. Change the value of the `sslProtocol` parameter to **TLSv1.2** and save the file.
4. Browse to the `JazzSM/profile/properties` path.
5. Open the `ssl.client.props` file in a flat-text editor.
6. Change the `com.ibm.ssl.protocol` parameter to `TLSv1.2` and save the file.
7. Run the following commands from the `TWA_home/wastools` directory to stop and restart the master domain manager:

```
./ stopWas.sh -direct -\user| wauser -password \password
./ startWas.sh -direct
```

8. Run the following commands from the `DWC_home/wastools` directory to stop and restart the Dynamic Workload Console:

```
./ stopWas.sh -direct -\user| DWUser -password \password
./ startWas.sh -direct
```

Converting default certificates

Procedure to extract and convert default certificates generated in your current version prior to upgrading.

About this task

If you are using default certificates, extract and convert them before you start the upgrade. Perform the following steps:

1. Set the HCL Workload Automation environment, as described in [Setting the environment variables on page 206](#).
2. To ensure the `keytool` and `openssl` commands start correctly on all operating systems, browse to the folder where the `keytool` and `openssl` commands are located and launch the commands as follows:

```
cd <TWS_DIR>/JavaExt/jre/jre/bin

./keytool -importkeystore -srckeystore TWSServerKeyFile.jks -destkeystore
<path_of_extracted_certs>/server.p12 -deststoretype pkcs12

cd <TWS_DIR>/tmpOpenSSL64/1.1/bin/openssl

./openssl pkcs12 -in <path_of_extracted_certs>/server.p12 -out
<path_of_extracted_certs>/tls.tot
```

The location of the `TWSServerKeyFile.jks` varies depending on the HCL Workload Automation version you have currently installed, as follows:

versions 9.5 and later

`TWA_DATA_DIR/usr/servers/engineServer/resources/security`

versions 9.4 and earlier

TWA_home/WAS/TWSPProfile/etc

3. Open the `tls.tot` file with any text editor.
4. From the `tls.tot` file, copy the private key to a new file named `tls.key`.

The `tls.key` file must be structured as follows:

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
<private_key>
-----END ENCRYPTED PRIVATE KEY-----
```



Note: Insert a carriage return after each key, so that an empty line is inserted after each key.

5. From the `tls.tot` file, copy the public key to a new file named `tls.crt`.

The `tls.crt` file must be structured as follows:

```
-----BEGIN CERTIFICATE-----
<public_key>
-----END CERTIFICATE-----
```



Note: Insert a carriage return after each key, so that an empty line is inserted after each key.

6. Copy the contents of the `tls.crt` file into a new file named `ca.crt`. If you want to upgrade a dynamic domain manager, also copy the contents of the `tls.crt` file into another new file named `jwt.crt`.
7. Create a file named `tls.sth` containing the passphrase you have specified for creating the `.p12` certificate in [step 2 on page 313](#), encoded in `base64` format. To create the `tls.sth` file, use the following command:

```
./secure -password your_password -base64 e -out
<path_of_extracted_certs>/tls.sth
```

If you are using a version earlier than 10.x, you can find the `secure` script in the installation package of the 10.2.5 version you are upgrading to. You can launch the script from one of the following paths:

master domain manager and agent

```
<10.2.5_extracted_image_dir>/TWS/<interp>/Tivoli_LWA_<interp>/TWS/bin
```

Dynamic Workload Console

```
<10.2.5_extracted_image_dir>/DWC/<interp>/bin
```

where

<interp>

is the operating system you are installing on

As an alternative, you can use the following command on UNIX workstations:

```
echo -n "passwordToEncode" | base64 >> tls.sth
```

- Browse to the GSKit folder and extract the client certificates from the `TWA_DATA_DIR/ssl/GSKit` folder by running the following commands, depending on the HCL Workload Automation version you have currently installed:

```
cd <TWS_DIR>/tmpGSKit64/8/bin
```

versions 9.5 and later

```
./gsk8capicmd_64 -cert -extract -db <TWA_DATA_DIR>/ssl/GSKit/TWSClietKeyStore.kdb  
-stashed -label client -target <path_of_extracted_certs>/client.crt
```

versions 9.4 and earlier

```
./gsk8capicmd_64 -cert -extract -db <TWS_DIR>/ssl/GSKit/TWSClietKeyStore.kdb  
-stashed -label client -target <path_of_extracted_certs>/client.crt
```

- Create a folder named `additionalCAs` in the folder where you extracted the certificates and move the `client.crt` file created in [step 8 on page 315](#) to the `additionalCAs` folder.
- Insert the `client.crt` in the `additionalCAs` folder when providing the certificates to the installation script with the **sslkeysfolder** parameter.
- Assign the correct permissions (755) and ownerships to extracted certificates, as follows:

```
chmod -R 755 <path_of_extracted_certs>
```

Results

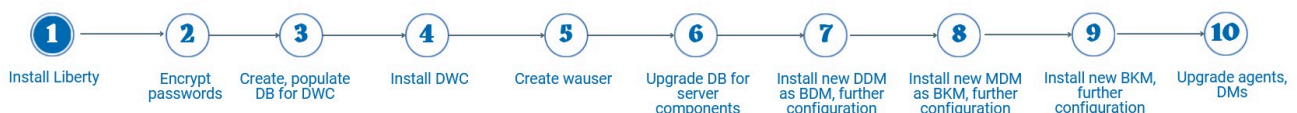
You have now extracted and converted your certificates for use with version 10.2.5.

What to do next

You can now upgrade WebSphere Application Server Liberty, as described in [Installing WebSphere Application Server Liberty on page 315](#). When upgrading HCL Workload Automation components in upcoming steps, provide the path to the folder where you extracted the certificates using the **sslkeysfolder** parameter when running the installation scripts. For more information about the installation scripts, see [Reference on page 427](#).

Installing WebSphere Application Server Liberty

Installing WebSphere Application Server Liberty to the latest supported version. This is an optional step you might want to perform before you upgrade the Dynamic Workload Console and the server components.



Before you begin

On AIX and Linux workstations, ensure you permanently set the **ulimit** parameter as follows:

- data segment process (option **-d**) = unlimited
- file size (option **-f**) = unlimited
- max user processes (option **-u**) = >260000 up to unlimited
- open files (option **-n**) = >100000 up to unlimited

- max memory size (option **-m**) = `unlimited`
- stack size (option **-s**) = `>33000 up to unlimited`

On the master domain manager, these settings must be applied to:

- `root`
- the HCL Workload Automation administrative user

On the Dynamic Workload Console, these settings must be applied to:

- `root`
- the Dynamic Workload Console installation user (if this user is different from root)

Ensure that your system meets the operating system requirements. For more information, see Open Liberty detailed system requirements.

About this task

You can quickly install Open Liberty by extracting an archive file on all supported platforms.

If you already have WebSphere Application Server Liberty Base installed, you can use it with HCL Workload Automation, otherwise you can install Open Liberty, as described below.

If you want to move from WebSphere Application Server Liberty Base to Open Liberty, see the topic about moving from WebSphere Application Server Liberty Base to Open Liberty in *Administration Guide*.

Install Open Liberty on all of the following workstations, which comprise a typical installation:

- master domain manager
- backup domain manager
- two Dynamic Workload Console installations on two separate workstations

If you plan to install a dynamic domain manager and its backup, these components require a separate Open Liberty installation.

On UNIX workstations, you can install Open Liberty using a user of your choice. In this case, assign the HCL Workload Automation administrative user read and write access to the Open Liberty installation directory.

To install Open Liberty, perform the following steps:

1. Find out which version of Open Liberty is required, by checking the required version of the Application server in the **Supported Software Report**, available in Product Requirements.
2. Download Open Liberty from [Get started with Open Liberty](#). Download the package named **All GA Features**
3. Perform one of the following actions:

- a. Extract Open Liberty using the root user:

On Windows operating systems

```
unzip <openliberty_download_dir>\openliberty-<version>.zip
-d <install_dir>
```

On UNIX operating systems

```
unzip <openliberty_download_dir>/openliberty-<version>.zip
-d <install_dir>
```

- b. Run the following command to assign permissions:

```
chmod 755 -R "wlp_directory"
```

OR

Extract Open Liberty using the user who is going to install the product, as follows:

```
su - "wuser"
unzip
```

where:

<openliberty_download_dir>

The directory where you downloaded Open Liberty.

install_dir

The directory where you want to install Open Liberty.



Note: Install the new Open Liberty in the exact location of the previous WebSphere Application Server Liberty Base installation.

4. Ensure the HCL Workload Automation administrative user has the rights to run Open Liberty and full access to the installation directory. If Open Liberty is shared between the master domain manager and the Dynamic Workload Console, ensure also the Dynamic Workload Console user has the same rights.

Results

You have now successfully installed Open Liberty.

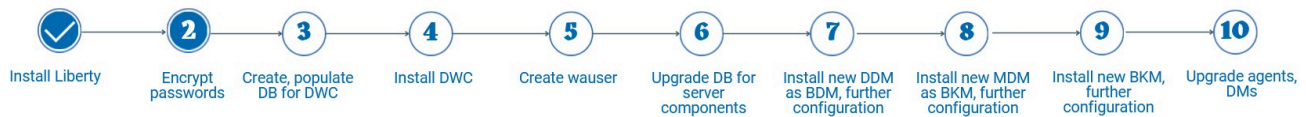
What to do next

You can now proceed to [Encrypting passwords \(optional\) on page 317](#) or to [Creating and populating the database for the Dynamic Workload Console on page 319](#).

Encrypting passwords (optional)

How to encrypt passwords required by the installation process

About this task



You can optionally encrypt the passwords that you will use while installing, upgrading, and managing HCL Workload Automation. The secure command uses the AES method and prints the encrypted password to the screen or saves it to a file.



Note: It is important you understand the limits to the protection that this method provides. The custom passphrase you use to encrypt the passwords is stored in clear format in the `passphrase_variables.xml` file, stored in `configureDropin`. To fully understand the implications of this method, it is recommended you read the information provided by Open Liberty at the link [Password encryption limitations](#).

You can perform a typical procedure, which uses a custom passphrase, as described in the following scenario. For more information about all secure arguments and default values, see [Optional password encryption - secure script on page 427](#).

Encrypting the password

1. Browse to the folder where the secure command is located:
 - Before the installation, the command is located in the product image directory, `<image_directory>/TWS/<op_sys>/Tivoli_LWA_<op_sys>/TWS/bin`
 - After the installation, the command is located in `TWA_home/TWS/bin`
2. Depending on your operating system, encrypt the password as follows:

Windows operating systems

```
secure -password password -passphrase passphrase
```

UNIX operating systems

```
./secure -password password -passphrase passphrase
```

z/OS operating systems

```
./secure -password password -passphrase passphrase
```

where

-password

Specifies the password to be encrypted.

-passphrase

Specifies the custom passphrase that is used to generate the key with which the command encrypts the password. If you set this parameter, inform the user who installs HCL Workload Automation that they must define the **SECUREWRAP_PASSPHRASE** environment variable in the same shell from which they run the installation command, and set it to the same value as the **passphrase** parameter. On Windows operating systems, the passphrase must be at least 8 characters long. This argument generates a

password which can be reused for all HCL Workload Automation components. This parameter is mutually exclusive with the `-useaeskeystore` on page 429 parameter, which generates a password which can be decrypted only on the local workstation and not reused for other components.

3. Provide both the encrypted password and custom passphrase to the user in charge of installing HCL Workload Automation. You can use encrypted passwords only in association with the specific passphrase used to encrypt them.

Installing with the encrypted password

The user in charge of installing HCL Workload Automation must set the **SECUREWRAP_PASSPHRASE** environment variable by performing the following steps:

1. Open a brand new shell session.
2. Ensure that no value is set for the **SECUREWRAP_PASSPHRASE** environment variable.
3. Define the **SECUREWRAP_PASSPHRASE** environment variable and set it to the passphrase defined by the user who ran the secure command, as follows:

```
SECUREWRAP_PASSPHRASE=<passphrase>
```

You can use encrypted passwords only in association with the specific passphrase used to encrypt them.

4. In the same shell session, provide the encrypted passwords when running any command that uses a password. An encrypted password looks like the following example:

```
{aes}AFC3jj9cR0YyqR+3CONBzVi8deLb2Bossb9GGroh8UmDPGikIkzXZzid3nzY0IhnSg=
```

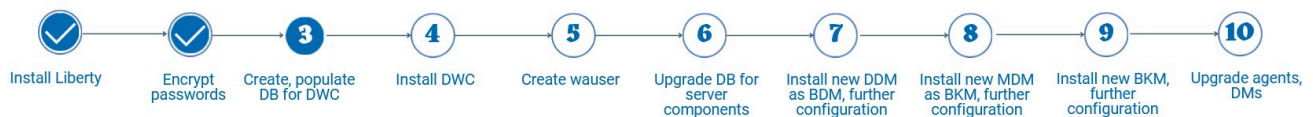
What to do next

You can now proceed to [Creating and populating the database for the Dynamic Workload Console](#) on page 319.

Creating and populating the database for the Dynamic Workload Console

Create and populate the database for the Dynamic Workload Console

About this task



If you are currently using Derby, you need to install a supported database and migrate your data. This is necessary because Derby is no longer supported as of version 10.2.3. For more information, see [Connecting the Dynamic Workload Console to a new node or database](#) on page 237.

If you are using a database other than Derby, create and populate the database tables for the Dynamic Workload Console by following the procedure appropriate for your RDBMS:

- [Creating and populating the database for DB2 for the Dynamic Workload Console on page 320](#)
- [Creating and populating the database for DB2 for z/OS for the Dynamic Workload Console on page 322](#)
- [Creating and populating the database for PostgreSQL for the Dynamic Workload Console on page 326](#)
- [Creating the database for Oracle and Amazon RDS for Oracle for the Dynamic Workload Console on page 328](#)
- [Creating and populating the database for MSSQL for the Dynamic Workload Console on page 330](#)
- [Creating and populating the database for MSSQL cloud-based databases for the Dynamic Workload Console on page 332](#)

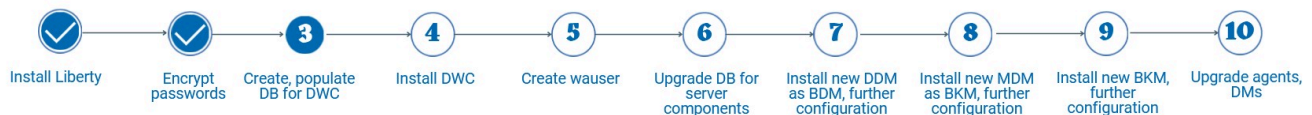
Creating and populating the database for DB2 for the Dynamic Workload Console

Instructions for creating and populating the Dynamic Workload Console database for DB2

Before you begin

Ensure a DB2 database is installed.

About this task



You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in the section about FAQ - Database customizations in Planning and Installation.

DB2 requires a specific procedure in which you first create the database and then create and populate the database tables. To simplify the database creation, a customized SQL file named `create_database.sql` is provided containing the specifics for creating the Dynamic Workload Console database. The database administrator can use this file to create the database. After the database has been created, you can proceed to create and populate the database tables.

You can optionally configure DB2 in SSL mode on UNIX operating systems by specifying the `sslkeysfolder` and `sslpassword` parameters when you run the `configureDb` command. For more information, see the topic about using certificates when DB2 or PostgreSQL is in SSL mode in *HCL Workload Automation: Planning and Installation*.

You can run the `configureDb` command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

Default values are stored in the `configureDb.properties` file, located in `image_location`. If you need to modify any of the default values, edit the `configureDb.properties` file, but do not modify the `configureDb.template` file located in the same path.

To create and populate the Dynamic Workload Console database and schema for DB2, perform the following steps:

1. On the workstation where you plan to install the Dynamic Workload Console, extract the Dynamic Workload Console package to a directory of your choice.
2. Browse to the *image_location/DWC_interp_name/tools* path.
3. Edit the *create_database.sql* file by replacing the default value for the database name (**DWC**) with the name you intend to use.
4. Provide the *create_database.sql* file to the DB2 administrator to run on the DB2 database.

The following command creates the Dynamic Workload Console database:

```
db2 -tvf file_location>/create_database.sql
```

5. Instruct the DB2 administrator to create the DB2 user on the server hosting the DB2 database. You will then specify this user with the `dbuser` parameter when creating and populating the database with the `configureDb` command on the Dynamic Workload Console. When you run the `configureDb` command, this user is automatically granted access to the Dynamic Workload Console tables on the database server.
6. On the server where you plan to install the Dynamic Workload Console, browse to *image_location/DWC_interp_name*.
7. Type the following command to create and populate the Dynamic Workload Console database tables with typical settings:

On Windows operating systems

```
cscript configureDb.vbs --rdbmstype DB2 --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype DB2 --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
```

where:

--rdbmstype

The database vendor.

--dbhostname *db_hostname*

The host name or IP address of database server.

--dbport *db_port*

The port of the database server.

--dbname *db_name*

The name of the Dynamic Workload Console database.

--dbuser *db_user*

The database user you must create before running the `configureDb` command. When you run the `configureDb` command, this user is automatically granted access to the Dynamic Workload Console tables on the database server.

--dbadminuser *db_admin_user*

The database administrator user that creates the Dynamic Workload Console schema objects on the database server.

--dbadminuserpw *db_admin_password*

The password of the DB administrator user that creates the Dynamic Workload Console schema objects on the database server. Special characters are not supported.



Note: The following parameters specified with the configureDb command are also required when installing the Dynamic Workload Console and their values must be the same:

- **--rdbmstype**
- **--dbhostname**
- **--dbport**
- **--dbname**
- **--dbuser**

Results

You have now successfully created and populated the Dynamic Workload Console database.

For more information about all parameters and supported values of the configureDb command, see [Database configuration - configureDb script on page 430](#).

What to do next

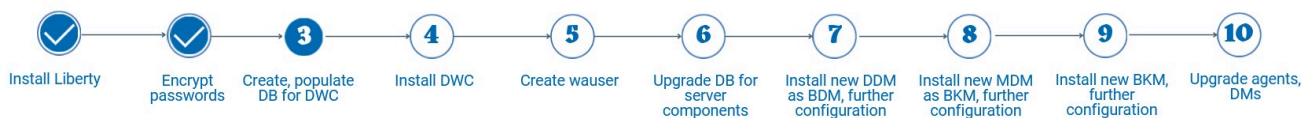
You can now proceed to [Installing the Dynamic Workload Console on page 334](#).

Creating and populating the database for DB2 for z/OS for the Dynamic Workload Console

Instructions for creating and populating the database for DB2 for z/OS for Dynamic Workload Console

Before you begin

Ensure a DB2 for z/OS database is installed.

About this task

You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in the section about FAQ - Database customizations in *HCL Workload Automation: Planning and Installation*.

DB2 for z/OS requires a specific procedure in which you first create the database and then create and populate the database tables. To simplify the database creation, a sample JCL named `EQQINDWC` is provided with Package `HWAZ_950_APAR_HC00001` containing the specifics for creating the Dynamic Workload Console database. The database administrator can use this file to create the database. After the database has been created, you can proceed to create and populate the database tables.



Note: If you use Db2 for z/OS with the Dynamic Workload Console version 10.2.4 or later, transfer the drivers in binary mode from the directory where you installed Db2 for z/OS to a directory of your choice. When you run the `configuredb` or `dwcinst` script, set the directory you chose in the `dbdriverspath` parameter.

You can run the **configureDb** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

If you need to modify any of the default values, edit the `configureDb<database_vendor>.properties` file, but do not modify the `configureDb<database_vendor>.template` file located in the same path.

To create and populate the Dynamic Workload Console database and schema for DB2 for z/OS, perform the following steps:

1. From the `SEQQSAMP` library, edit the `EQQINDWC` sample JCL as required.



Note: The `EQQINDWC` sample JCL is provided with the Package `HWAZ_950_APAR_HC00001`. If you did not install this APAR, create a JCL named `EQQINDWC` that looks like the following example:

```
//JOB CARD
//*****
//*
//* SECURITY CLASSIFICATION:
//* Licensed Materials - Property of HCL 5698-T08
//* Copyright HCL Technologies Ltd. 2020 All rights reserved.
//* US Government Users Restricted Rights - Use, duplication
//* or disclosure restricted by GSA ADP Schedule Contract
//*
//*
//* CREATES DB2 STORAGE GROUP AND DATABASE for DWC
//* NOTE1: You must tailor this JCL sample to conform to
//* installation standards defined at your location.
//* - Add a JOB card
//* - Change following DB/2 values according to your
//* current environment:
//* - DSN.V11R1M0.SDSNLOAD DB/2 library
//* - DSN111.RUNLIB.LOAD DB/2 run library
//* - DBB1 DB/2 system name
//* - DSNTIA11 DB/2 DSNTIAD plan name
//* - volname volume name
//* - catname catalog name
//* - Change all the occurrences of
//* TWSSDWC if you need a storage group with a different name
//*
//* Flag Reason Rlse Date Origin Flag Description
```



```
//* ----- */
//* $EGE=PH22448 950 200121 ZLIB: DB2 on zLiberty */
//* $ETA=PH53936 101 220418 MR: EQQINDWC MEMBER OF SEQQSAMP FOR */
//* CREATION OF DB2 DATABASE FOR */
//* DWCFails FOR DB2 V12R1M504 OR */
//* higher levels */
/*****/
//EQQINDWC EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DISP=SHR,DSN=DSN.V11R1M0.SDSNLOAD
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
    DSN SYSTEM(DBB1)
    RUN PROGRAM(DSNTIAD) PLAN(DSNTIA11) LIB('DSN111.RUNLIB.LOAD')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
SET CURRENT APPLICATION COMPATIBILITY = 'V10R1';
CREATE STOGROUP TWSSDWC VOLUMES(volname) VCAT catname;
CREATE DATABASE DWC
BUFFERPOOL BP0
INDEXBP BP16K0
STOGROUP TWSSDWC
CCSID UNICODE;
COMMIT;
```

2. Instruct the DB2 for z/OS administrator to create the DB2 for z/OS user on the server hosting the DB2 for z/OS database. You will then specify this user with the `dbuser` parameter when creating and populating the database with the `configureDb` command on the Dynamic Workload Console. When you run the `configureDb` command, this user is automatically granted access to the Dynamic Workload Console tables on the database server.
3. On the server where you plan to install the Dynamic Workload Console, browse to the directory where you extracted the Dynamic Workload Console image.
4. Type the following command to create and populate the Dynamic Workload Console database tables with typical settings:

On Windows operating systems

```
cscript configureDb.vbs --rdbmstype DB2Z --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
--zlocationname zOS_location_containing_db --zbufferpoolname buffer_pool_in_zOS_location
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype DB2Z --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
--zlocationname zOS_location_containing_db --zbufferpoolname buffer_pool_in_zOS_location
```

On z/OS operating systems

```
./configureDb.sh --rdbmstype DB2Z --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
--zlocationname zOS_location_containing_db --zbufferpoolname buffer_pool_in_zOS_location
```


where:

--rdbmstype

The database vendor.

--dbhostname *db_hostname*

The host name or IP address of database server.

--dbport *db_port*

The port of the database server.

--dbname *db_name*

The name of the Dynamic Workload Console database.

--dbuser *db_user*

The database user you must create before running the `configureDb` command. When you run the `configureDb` command, this user is automatically granted access to the Dynamic Workload Console tables on the database server.

--dbadminuser *db_admin_user*

The database administrator user that creates the Dynamic Workload Console schema objects on the database server.

--dbadminuserpw *db_admin_password*

The password of the DB administrator user that creates the Dynamic Workload Console schema objects on the database server. Special characters are not supported.

--zlocationname *zos_location_containing_db*

The name of an already existing location in the z/OS environment that will contain the new database. The default value is LOC1.

--zbufferpoolname *buffer_pool_in_zos_location*

The name of an already existing buffer pool created in the location specified by `-zlocationname`. The default value is BP32K.



Note: The following parameters specified with the `configureDb` command are also required when installing the Dynamic Workload Console and their values must be the same:

- **--rdbmstype**
- **--dbhostname**
- **--dbport**
- **--dbname**



- `--dbuser`
- `--zlocationname`

Results

You have now successfully created and populated the Dynamic Workload Console database.

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

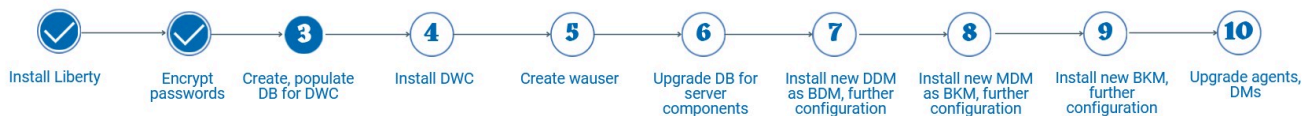
What to do next

You can now proceed to [Installing the Dynamic Workload Console on page 334](#).

Creating and populating the database for PostgreSQL for the Dynamic Workload Console

Instructions for creating and populating the Dynamic Workload Console database for PostgreSQL

About this task



You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in [FAQ - Database customizations on page 87](#).

You can run the `configureDb` command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

Default values are stored in the `configureDbPostgresql.properties` file, located in `image_location`. If you need to modify any of the default values, edit the `configureDbPostgresql.properties` file, but do not modify the `configureDbPostgresql.template` file located in the same path. For an example of a properties file, see [What is the content of a database properties file? on page 96](#).

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

To create and populate the Dynamic Workload Console database, perform the following steps:

1. Check the collation settings before proceeding, as described in [Incorrect collation settings in PostgreSQL database on page 409](#).
2. On the server where you plan to install the Dynamic Workload Console, extract the Dynamic Workload Console package to a directory of your choice.
3. Browse to the directory where you extracted the package.
4. Type the following command to populate the Dynamic Workload Console database with typical settings:

On Windows operating systems

```
cscript configureDb.vbs --rdbmstype POSTGRESQL
--dbuser db_user --dbpassword DB_password --dbhostname DB_hostname
--dbadminuser DB_administrator --dbadminuserpw DB_administrator_password
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype POSTGRESQL
--dbuser db_user --dbpassword DB_password --dbhostname DB_hostname
--dbadminuser DB_administrator --dbadminuserpw DB_administrator_password
```

where:

--rdbmstype

The database vendor.

--dbname *db_name*

The service name of the Dynamic Workload Console database.

--dbuser *db_user*

The user to be granted access to the Dynamic Workload Console tables on the database server.

--dbpassword *db_password*

The password for the user that has been granted access to the Dynamic Workload Console tables on the database server. Special characters are not supported.

--dbhostname *db_hostname*

The host name or IP address of database server.

--dbadminuser *db_admin_user*

The database administrator user that creates the Dynamic Workload Console schema objects on the database server.

--dbadminuserpw *db_admin_password*

The password of the DB administrator user that creates the Dynamic Workload Console schema objects on the database server. Special characters are not supported.



Note: The following parameters specified with the configureDb command are also required when installing the Dynamic Workload Console and their values must be the same:

- **rdbmstype**
- **dbhostname**
- **dbport**
- **dbname**
- **dbuser**
- **dbpassword**

Results

You have now successfully created and populated the Dynamic Workload Console database.

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

What to do next

You can now proceed to [Installing the Dynamic Workload Console on page 334](#).

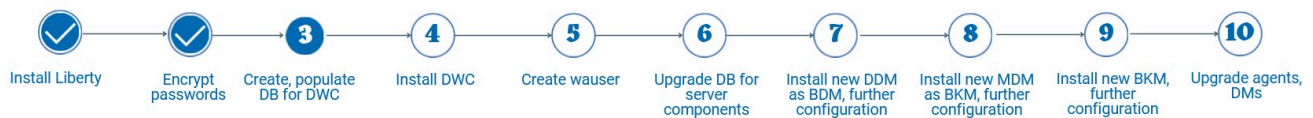
Creating the database for Oracle and Amazon RDS for Oracle for the Dynamic Workload Console

Instructions for creating and populating the Dynamic Workload Console database for Oracle and Amazon RDS for Oracle

Before you begin

Ensure the required tablespace for Dynamic Workload Console data has been already created on the Oracle database server which hosts the Dynamic Workload Console database.

About this task



You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in [FAQ - Database customizations on page 87](#).

You can run the `configureDb` command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

Default values are stored in the `configureDbOracle.properties` file, located in `image_location`. If you need to modify any of the default values, edit the `configureDbOracle.properties` file, but do not modify the `configureDbOracle.template` file located in the same path.

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

To create and populate the Dynamic Workload Console database, perform the following steps:

1. On the server where you plan to install the Dynamic Workload Console, extract the Dynamic Workload Console package to a directory of your choice.
2. Browse to the directory where you extracted the package.
3. Type the following command to populate the Dynamic Workload Console database with typical settings:

On Windows operating systems

```

cscript configureDb.vbs --rdbmstype ORACLE --dbname service_name
--dbuser db_user --dbpassword DB_password --dbhostname DB_hostname
  
```

```
--dbadminuser DB_administrator --dbadminuserpw DB_administrator_password
--iwststname USERS
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype ORACLE --dbname service_name
--dbuser db_user --dbpassword DB_password --dbhostname DB_hostname
--dbadminuser DB_administrator --dbadminuserpw DB_administrator_password
--iwststname USERS
```

where:

--rdbmstype

The database vendor.

--dbname db_name

The service name of the Dynamic Workload Console database.

--dbuser db_user

The user to be granted access to the Dynamic Workload Console tables on the database server.

--dbpassword db_password

The password for the user that has been granted access to the Dynamic Workload Console tables on the database server. Special characters are not supported.

--dbhostname db_hostname

The host name or IP address of database server.

--dbadminuserdb_admin_user

The database administrator user that creates the Dynamic Workload Console schema objects on the database server.

--dbadminuserpw db_admin_password

The password of the DB administrator user that creates the Dynamic Workload Console schema objects on the database server. Special characters are not supported.

--iwststname|tn table_space_name

The name of the tablespace for Dynamic Workload Console data. This parameter is required.



Note: The following parameters specified with the configureDb command are also required when installing the Dynamic Workload Console and their values must be the same:

- **rdbmstype**
- **dbhostname**
- **dbport**
- **dbname**



- **dbuser**
- **dbpassword**

Results

You have now successfully created and populated the Dynamic Workload Console database.

For more information about all parameters and supported values of the configureDb command, see [Database configuration - configureDb script on page 430](#).

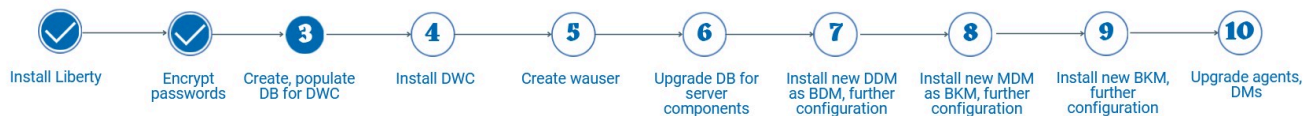
What to do next

You can now proceed to [Installing the Dynamic Workload Console on page 334](#).

Creating and populating the database for MSSQL for the Dynamic Workload Console

Instructions for creating and populating the Dynamic Workload Console database for MSSQL

About this task



You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in [FAQ - Database customizations on page 87](#).

You can run the configureDb command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. By default, MSSQL authentication is used. To modify the authentication type, see [How can I specify the authentication type when using an MSSQL database? on page 93](#).

For more information about all parameters and supported values of the configureDb command, see [Database configuration - configureDb script on page 430](#). If you need to modify any of the default values, edit the `configureDbMSSQL.properties` file, but do not modify the `configureDbMSSQL.template` file located in the same path. For an example of a properties file, see [What is the content of a database properties file? on page 96](#).

Default values are stored in the `configureDbMSSQL.properties` file, located in `image_location`.



Note: Only on Windows systems hosting an MSSQL database, the path hosting the tablespace must be existing before you run the `configureDb.vbs` command.

To create the Dynamic Workload Console database and schema, perform the following steps:

1. Only on Windows systems hosting an MSSQL database, create the path for hosting the following tablespace, if the path is not already existing:
 - TWS_DATA

2. Only on Windows systems hosting an MSSQL database, specify the path to the folder when running the `configureDb.vbs` command or when filling in the `configureDbMSSQL.properties` properties file with the following parameter:
 - `--iwstspath`
3. On the server where you plan to install the Dynamic Workload Console, extract the Dynamic Workload Console package to a directory of your choice.
4. To populate the Dynamic Workload Console database with typical settings, type the following command:

On Windows operating systems

```
cscript configureDb.vbs --rdbmstype MSSQL --dbname db_name
--dbhostname db_hostname --dbadminuser db_administrator
--dbadminuserpw db_administrator_password
--iwstspath DATA_tablespace_path
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype MSSQL --dbname db_name
--dbhostname db_hostname --dbadminuser db_administrator
--dbadminuserpw db_administrator_password
--iwstspath DATA_tablespace_path
```

where:

--rdbmstype

The database vendor.

--dbname *db_name*

The name of the Dynamic Workload Console database.

--dbhostname *db_hostname*

The host name or IP address of database server.

--dbadminuser *db_admin_user*

The database administrator user that creates the Dynamic Workload Console schema objects on the database server.

--dbadminuserpw *db_admin_password*

The password of the DB administrator user that creates the HCL Workload Automation schema objects on the database server. Special characters are not supported.

--iwstspath|-tp *table_space*

The path of the tablespace for HCL Workload Automation or Dynamic Workload Console data. This parameter is optional. The default value for all databases other than Oracle is:

For all operating systems, except z/OS

TWS_DATA

For z/OS operating system

TWSDATA

Only on Windows systems hosting an MSSQL database, ensure the folder for the tablespace is already existing before running the configureDb command and specify the path using this parameter. Specify the path using forward slashes (/), for example: `c : / <my_path> / TWS_DATA`.



Note: The following parameters specified with the configureDb command are also required when installing the Dynamic Workload Console and their values must be the same:

- **rdbmstype**
- **dbhostname**
- **dbport**
- **dbname**

When **--rdbmstype** is set to `MSSQL`, the default value is **sa**. To install a Dynamic Workload Console with a user different from **sa**, you must create a new user in `MSSQL` and grant all the required permissions before running the configureDb command.

Results

You have now successfully created and populated the Dynamic Workload Console database.

For more information about all parameters and supported values of the configureDb command, see [Database configuration - configureDb script on page 430](#).

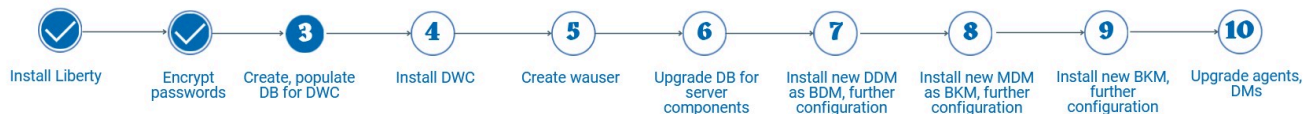
What to do next

You can now proceed to [Installing the Dynamic Workload Console on page 334](#).

Creating and populating the database for MSSQL cloud-based databases for the Dynamic Workload Console

Instructions for creating and populating the Dynamic Workload Console database for MSSQL cloud-based databases

About this task



MSSQL cloud-based databases include the following:

- Azure SQL
- Google Cloud SQL for SQL server
- Amazon RDS for MSSQL

You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in [FAQ - Database customizations on page 87](#).

You can run the `configureDb` command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#). If you need to modify any of the default values, edit the `configureDbMSSQL.properties` file, but do not modify the `configureDbMSSQL.template` file located in the same path. For an example of a properties file, see [What is the content of a database properties file? on page 96](#).

Default values are stored in the `configureDbMSSQL.properties` file, located in *image_location*.

To create the Dynamic Workload Console database and schema, perform the following steps:

1. Specify the path to the folder when running the `configureDb` command or when filling in the `configureDbMSSQL.properties` properties file with the following parameter:
 - `--iwstname PRIMARY`

You can optionally modify the `PRIMARY` default value when running the `configureDb` command.
2. On the server where you plan to install the Dynamic Workload Console, extract the Dynamic Workload Console package to a directory of your choice.
3. To populate the Dynamic Workload Console database with typical settings, type the following command:

On Windows operating systems

```
cscript configureDb.vbs --rdbmstype MSSQL --dbname db_name
--dbhostname db_hostname --dbadminuser db_administrator
--dbadminuserpw db_administrator_password
--iwstname DATA_tablespace_name
```

`iwstname DATA_tablespace_name`

The name of the tablespace for Dynamic Workload Console data. This parameter is required.



Note: The following parameters specified with the **`configureDb`** command are also required when installing the Dynamic Workload Console and their values must be the same:

- **`rdbmstype`**
- **`dbhostname`**
- **`dbport`**
- **`dbname`**
- **`dbuser`**

Results

You have now successfully created and populated the Dynamic Workload Console database.

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

What to do next

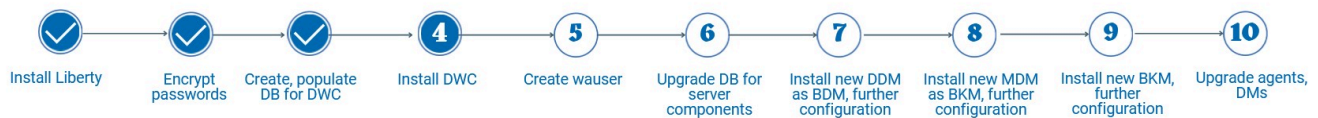
You can now proceed to [Installing the Dynamic Workload Console on page 334](#).

Installing the Dynamic Workload Console

Procedure for installing two Dynamic Workload Console servers on two separate nodes.

About this task

Figure 12. Install fresh Dynamic Workload Console



The procedure to perform a fresh installation is demonstrated through a typical scenario where two Dynamic Workload Console servers are installed on separate workstations, sharing the same remote database.

With Version 9.5, the Dynamic Workload Console is based on a new architectural foundation that does not include Jazz for Service Management nor Dashboard Application Services Hub, therefore, no direct upgrade procedure is supported, but you perform a fresh installation of the Dynamic Workload Console at version 9.5.0.x or 10.2.x.



Note: If you are installing the Dynamic Workload Console version 10.2.3 or later, the Federator is also automatically installed. This component enables you to monitor your objects through the Orchestration Monitor page of the Dynamic Workload Console. For detailed information about how to configure and use the Federator, see *Mirroring the z/OS current plan to enable the Orchestration Monitor* in the section about mirroring the z/OS current plan to enable the Orchestration Monitor in the *Dynamic Workload Console User's Guide*.

If you are currently using Derby, you need to install a supported database and migrate your data. This is necessary because Derby is no longer supported as of version 10.2.3. For more information, see [Connecting the Dynamic Workload Console to a new node or database on page 237](#).

In this scenario, the HCL Workload Automation administrator installs two Dynamic Workload Console instances on two separate workstations, sharing the same remote database. The HCL Workload Automation administrator performs the operations listed below on both workstations.

Certificates are now required when installing or upgrading HCL Workload Automation. You can no longer install nor upgrade HCL Workload Automation without securing your environment with certificates. The required certificates are:

- ca.crt
- tls.key
- tls.crt

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

Convert the certificates as described in [Converting default certificates on page 313](#) and copy them locally.

The HCL Workload Automation administrator installs the Dynamic Workload Console. The following information is required:

Table 18. Required information

Command parameter	Required information	Provided in..
Database information		
--rdbmstype	database type	Creating and populating the database for the Dynamic Workload Console on page 319
--dbhostname	database hostname	
--dbport	database port	
--dbname	database name	
--dbuser	database user name	
--dbpassword	database password	
WebSphere Application Server Liberty Base information		
--wlpdir	WebSphere Application Server Liberty Base installation directory	Installing WebSphere Application Server Liberty on page 315
Security information		
--sslkesfolder	location of converted certificates	Converting default certificates on page 313
--sslpassword	password of converted certificates	Converting default certificates on page 313

You can run the **dwcinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

Default values are stored in the `dwcinst.properties` file, located in the root directory of the installation image.

If you need to modify any of the default values, edit the `dwcinst.properties` file, but do not modify the `dwcinst.template` file located in the same path.

In a typical installation scenario, it is recommended you install the Dynamic Workload Console as a **non-root user** on UNIX systems and as a **local administrator** on Windows systems.

This user is automatically created by the installation process in the WebSphere Application Server Liberty Base repository. Ensure that the user has full access to the WebSphere Application Server Liberty Base installation directory.

To install the Dynamic Workload Console, perform the following steps:

1. Log in to the workstation where you plan to install the Dynamic Workload Console.
2. Download the installation images from [HCL Software](#).
3. Browse to the folder where the `dwcinst` command is located in `image_location/TWS/interp_name`.

4. Start the installation specifying a typical set of parameters:

On Windows operating systems

```
cscript dwcinst.vbs --acceptlicense yes --rdbmstype db_type
--user dwc_admin_user --password dwc_pwd --dbname db_name
--dbuser db_user --dbpassword db_pwd --dbhostname db_hostname
--dbport db_port --wlpdir Liberty_installation_dir\wlp
--sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
```

On UNIX operating systems

```
./dwcinst.sh --acceptlicense yes --rdbmstype db_type
--user dwc_admin_user --password dwc_pwd --dbname db_name
--dbuser db_user --dbpassword db_pwd --dbhostname db_hostname
--dbport db_port --wlpdir Liberty_installation_dir/wlp
--sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
```

where,

user *dwc_admin_user*

is the administrator of the Dynamic Workload Console. This user is added to the group of the Dynamic Workload Console administrators at installation time. You can use this account to log in to the Dynamic Workload Console and manage your environment.

password *dwc_pwd*

is the password of the Dynamic Workload Console user.

On Windows operating systems

Supported characters for the password are alphanumeric, dash (-), underscore (_) characters, and ()|?*~+. @!^

On UNIX operating systems

Supported characters for the password are any alphanumeric, dash (-), underscore (_) characters, and ()|?*~+.

Results

You have now successfully installed the Dynamic Workload Console.

For more information about all **dwcinst** parameters and default values, see [Dynamic Workload Console installation - dwcinst script on page 456](#).

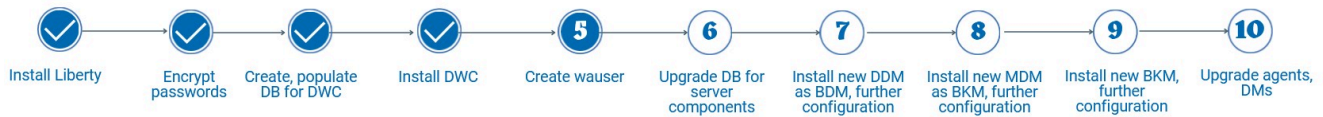
What to do next

If you had previously exported the Dynamic Workload Console, as described in [Connecting the Dynamic Workload Console to a new node or database on page 237](#), you can now import them in the new Dynamic Workload Console from the **Administration > Manage Settings** menu. If you have a high availability configuration, import the settings on one node.

You can now proceed to [Creating the HCL Workload Automation administrative user on page 337](#).

Creating the HCL Workload Automation administrative user

Instructions to create the HCL Workload Automation administrative user



HCL Workload Automation administrative user

The HCL Workload Automation administrator creates the administrative user (**wauser**). The administrative user is the user for which the product will be installed in the subsequent steps. This implies that this user has full access to all scheduling objects.

The user name can contain alphanumeric, dash (-), and underscore (_) characters; it cannot contain national characters. The first character of the user name must be a letter.

The following considerations apply:

On Windows operating systems:

- If this user account does not already exist, it is automatically created at installation time.
- If installing on a Windows™ server in a domain, do not define a domain and local ID with the same user name.
- If you specify a domain user, define the name as *domain_name\user_name*.
- If you specify a local user, define the name as *system_name\user_name*. Type and confirm the password.

On UNIX and Linux operating systems:

This user account must be created manually before running the installation and must be enabled to login to the machine where the master domain manager is going to be installed. Create a user with a home directory and group. Use the appropriate UNIX and Linux operating system commands to create the user.

Important: Group names that contain a "/" (forward slash) character can cause permissions to not be set correctly. When HCL Workload Automation retrieves credentials from WebSphere Application Server Liberty, it parses the returned list of groups names assuming they are saved in the format `<realm_name>/<group_name>`. If the group name, the realm name, or both contain a "/" character, the parsing fails.

You can also install HCL Workload Automation using a user different from the root user. This installation method is known as **no-root installation** and applies to all HCL Workload Automation components. Note that if you choose this installation method, only the user who performs the installation can use HCL Workload Automation. For this reason, the typical installation scenario described in this section uses the root user.

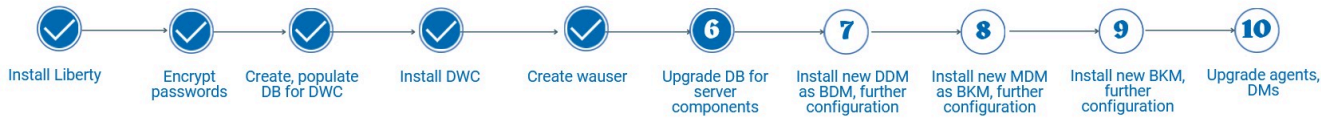
For more information, see [HCL Workload Automation user management on page 49](#).

What to do next

You can now proceed to [Upgrading the database for the server components on page 338](#).

Upgrading the database for the server components

Upgrade the master domain manager database tables before upgrading the server



components.

Before you begin



Note: Before upgrading the database schema, ensure you have created a backup. Refer to the documentation related to your RDBMS for information about the backup procedure.



Note: If you are using a PostgreSQL database, check the collation settings before proceeding, as described in [Incorrect collation settings in PostgreSQL database on page 409](#).

Ensure you have acquired information about the HCL Workload Automation tablespaces that were specified when the database tables were created and populated the first time. If values different from the default values were used, then your database administrator must provide them for this upgrade procedure. If default values were used, then they do not need to be specified during the upgrade procedure. The default values for the HCL Workload Automation data, log, and plan tablespaces are as follows:

- **--iwstname** TWS_DATA
 - For Oracle only, the default is `USERS`
- **--iwslogtsname** TWS_LOG
 - For Oracle only, the default is `USERS`
- **--iwsplantsname** TWS_PLAN
 - For Oracle only, the default is `USERS`

For more information about all parameters and supported values of the `configureDb` command, see [Database configuration - configureDb script on page 430](#).

About this task

You can run the `configureDb` command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

The script creates an SQL file with all the statements needed to upgrade the HCL Workload Automation database schema to the latest version and, by default, automatically applies it.

Default values are stored in the `configureDb<database_vendor>.properties` file, located in `image_location/TWS/interp_name`. For an example of a properties file, see [What is the content of a database properties file? on page 96](#).

If you need to modify any of the default values, edit the `configureDb<database_vendor>.properties` file, but do not modify the `configureDb<database_vendor>.template` file located in the same path.

To upgrade the HCL Workload Automation database schema, perform the following steps:

1. On the workstation where you plan to install the new backup master domain manager or backup dynamic domain manager, extract the HCL Workload Automation package at the latest version to a directory of your choice.
2. Browse to the `image_location/TWS/interp_name` path.
3. Type the following command to upgrade the HCL Workload Automation database schema to the latest version. Ensure that you use the same database administrator credentials you used when the HCL Workload Automation database schema objects were created. The new backup master domain manager or backup dynamic domain manager is configured to point to the existing database instance.

On Windows operating systems

```
cscript configureDb.vbs --rdbmstype db_vendor --dbhostname db_hostname --dbport db_port
--dbname db_name --dbuser db_user --componenttype server_component
--dbadminuser db_administrator --dbadminuserpw db_administrator_password
--iwsname tablespace_data --iwslogtsname tablespace_log --iwsplantsname tablespace_plan
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype db_vendor --dbhostname db_hostname --dbport db_port
--dbname db_name --dbuser db_user --componenttype server_component
--dbadminuser db_administrator --dbadminuserpw db_administrator_password
--iwsname tablespace_data --iwslogtsname tablespace_log --iwsplantsname tablespace_plan
```

where:

--rdbmstype

The database vendor.

--dbhostname db_hostname

The host name or IP address of database server.

--dbport db_port

The port of the database server.

--dbname db_name

The name of the HCL Workload Automation database.

--dbuser db_user

The user that has been granted access to the HCL Workload Automation tables on the database server.

--dbpassword *db_password*

The password for the user that has been granted access to the HCL Workload Automation tables on the database server. Special characters are not supported.

--dbadminuser *db_admin_user*

The database administrator user that creates the HCL Workload Automation schema objects on the database server.

--dbadminuserpw *db_admin_password*

The password of the DB administrator user that creates the HCL Workload Automation schema objects on the database server. Special characters are not supported.

--componenttype **MDM | DDM**

The HCL Workload Automation component for which the database is installed. This parameter is optional. Supported values are:

MDM

master domain manager.

DDM

dynamic domain manager.

--iwstsname *tablespace_data*

The name of the tablespace for HCL Workload Automation data. The default value for all supported RDBMS is TWS_DATA, with the exception of Oracle where the default is USERS.

--iwslogtsname *tablespace_log*

The name of the tablespace for the HCL Workload Automation log. The default value for all supported RDBMS is TWS_LOG, with the exception of Oracle where the default is USERS.

--iwsplantsname *db_port*

The name of the tablespace for the HCL Workload Automation plan. The default value for all supported RDBMS is TWS_PLAN, with the exception of Oracle where the default is USERS.

--auth_type *db_name*

The MSSQL authentication mode. The default is SQLSERVER which uses native SQL authentication.

You can optionally point the backup master domain manager to different database residing on the same workstation. For more information, see the topic about Connecting the master domain manager to a new database in *Administration Guide*.



Note: The following parameters specified with the `configureDb` command are also required when you upgrade the server components with the `serverinst` command and their values must be the same:

- **rdbmstype**
- **dbhostname**



- **dbport**
- **dbname**
- **dbuser**
- **dbpassword**

Results

You have now successfully upgraded the database schema for the HCL Workload Automation database.

What to do next

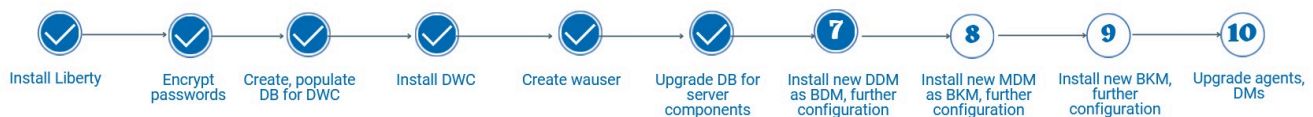
You can now proceed to [Installing a new dynamic domain manager configured as a backup on page 341](#) or to [Installing the new master domain manager configured as a backup on page 348](#).

Installing a new dynamic domain manager configured as a backup

Install a new dynamic domain manager configured as a backup and link it to your current network. Then switch it to become the new dynamic domain manager.

About this task

This is a parallel upgrade procedure that installs a fresh dynamic domain manager configured as backup. The dynamic domain manager configured as a backup points to your existing HCL Workload Automation database and then later becomes your new dynamic domain manager.



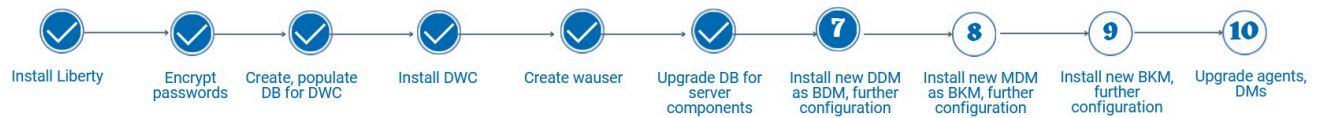
This section describes how to upgrade the dynamic components (dynamic domain manager and its backup). For details about the supported versions from which you can upgrade, see the [HCL Workload Automation Release Notes](#).

Perform the following steps:

1. [Installing a new dynamic domain manager configured as a backup on page 341](#)
2. [Switching the dynamic domain manager to the new dynamic domain manager configured as backup on page 344](#)
3. [Installing a new backup dynamic domain manager on page 345](#) to replace the backup dynamic domain manager which you have switched to become the current dynamic domain manager.
4. [Switching back to the old dynamic domain manager \(optional\) on page 348](#)

Installing a new dynamic domain manager configured as a backup

Procedure for installing a dynamic domain manager configured as a backup



Install a new dynamic domain manager at the latest product version level configured as the new backup dynamic domain manager by running the serverinst script.

The procedure to install the dynamic domain manager and backup dynamic domain manager is exactly the same, with the difference that it is performed on two different workstations and that each installation points to its local Open Liberty installation. HCL Workload Automation determines whether or not a dynamic domain manager is already present in the environment and proceeds to install a dynamic domain manager or backup dynamic domain manager accordingly.

The HCL Workload Automation administrator installs the dynamic domain manager as the backup. The following information is required:

Table 19. Required information

Command parameter	Information type	Provided in...
Database information		
--rdbmstype	database type	Upgrading the database for the server components on page 338
--dbhostname	database hostname	
--dbport	database port	
--dbname	database name	
--dbuser	database user name	
--dbpassword	database password	
HCL Workload Automation information		
--wauser	HCL Workload Automation administrative user name	Creating the HCL Workload Automation administrative user on page 337
--wapassword	HCL Workload Automation administrative user password	
WebSphere Application Server Liberty Base information		
--wlpdir	WebSphere Application Server Liberty Base installation directory	Installing WebSphere Application Server Liberty on page 315
Security information		

Table 19. Required information

(continued)

--sslkeyfolder	location of converted certificates	Converting default certificates on page 313
--sslpassword	password of converted certificates	Converting default certificates on page 313

Before starting the installation, ensure the following steps have been completed:

1. [Converting default certificates on page 313](#). Because you are installing a dynamic domain manager, also copy locally the `jwt.crt` file created in the conversion procedure.
2. [Installing WebSphere Application Server Liberty on page 315](#) on the workstation where you plan to install the dynamic domain manager and on the workstation where you plan to install the backup dynamic domain manager.
3. [Encrypting passwords \(optional\) on page 317](#)
4. [Upgrading the database for the server components on page 338](#)
5. [Creating the HCL Workload Automation administrative user on page 337](#)

You can run the **serverinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. For more information about all **serverinst** parameters and default values, see [Server components installation - serverinst script on page 442](#).

A properties file named `serverinst.properties` is available if you do not want to type parameters in the command line. This is especially useful if you need to specify many parameters or if you want to reuse the file for several installations. The file is located in `image_location/TWS/interp_name`.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

If you need to modify any of the default values, edit the `serverinst.properties` file, but do not modify the `serverinst.template` file located in the same path.

To install the dynamic domain manager as a backup, perform the following steps:

1. Log in to the workstation where you plan to install.
2. Browse to the folder where the `serverinst` command is located:

On Windows operating systems

`image_location\TWS\interp_name`

On UNIX operating systems

`image_location/TWS/interp_name`

3. Start the installation specifying a typical set of parameters. In this case, default values are used for all remaining parameters:

On Windows operating systems

```
cscript serverinst.vbs --acceptlicense yes --rdbmstype db_type
--dbhostname db_hostname --dbport db_port --dbname db_name
--dbuser db_user --dbpassword db_password --wauser wa_user
--wapassword wa_password --componenttype DDM --domain domain_name
--master mdm_name --mdmbrokerhostname mdm_broker_host_name
--sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
--mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir\wlp
```

On UNIX operating systems

```
./serverinst.sh --acceptlicense yes --rdbmstype db_type
--dbhostname db_hostname --dbport db_port --dbname db_name
--dbuser db_user --dbpassword db_password --wauser wa_user
--wapassword wa_password --componenttype DDM --domain domain_name
--master mdm_name --mdmbrokerhostname mdm_broker_host_name
--sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
--mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir/wlp
```

4. Distribute the Symphony file to the new dynamic domain manager configured as backup:
 - a. Ensure that the **optman cf** option is set to *all*.
 - b. To distribute the Symphony file to the new dynamic domain manager configured as backup, run `JnextPlan -for 0000` or wait until the end of the production plan.
 - c. Restore the previous setting of the **optman cf** option, if you previously modified the value.

You have now successfully installed the backup dynamic domain manager at the new product version level.

For more information about all **serverinst** parameters and default values, see [Server components installation - serverinst script on page 442](#).

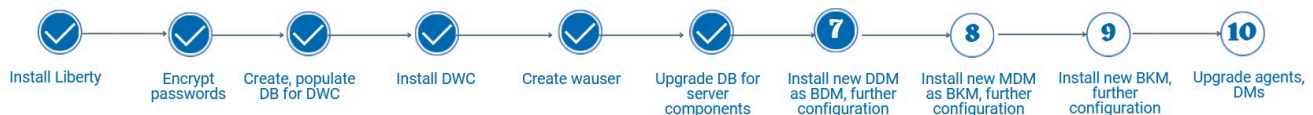
What to do next

You can now proceed to [Switching the dynamic domain manager to the new dynamic domain manager configured as backup on page 344](#).

Switching the dynamic domain manager to the new dynamic domain manager configured as backup

Switch the old dynamic domain manager to become a backup dynamic domain manager. As a result, the backup dynamic domain manager you installed in the previous step, becomes the current dynamic domain manager.

About this task



Switch to your new dynamic domain manager configured as backup, so that it becomes your current dynamic domain manager, by completing the following steps:

1. Stop the workload broker server on the dynamic domain manager at the previous product version level, by running the following command:

On Windows operating systems

```
stopBrokerApplication.bat
-user username -password password
[-port portnumber]
```

On UNIX and Linux operating systems

```
stopBrokerApplication.sh
-user username -password password
[-port portnumber]
```

where *username* and *password* are the values specified during the dynamic domain manager installation. The parameter *portnumber* is optional, if it is not specified, the default is used.

2. Switch the dynamic domain manager to its backup workstation. Use either the Dynamic Workload Console or run the command:

```
conman
switchmgr dyn_dom;new_mgr_cpu
```

where *dyn_dom* is the domain where you installed the backup dynamic domain manager and the *new_mgr_cpu* is the backup dynamic domain manager workstation name.

3. From the new current dynamic domain manager, unlink the old dynamic domain manager workstation:

```
conman "unlink old_ddm_wks"
```

where *old_ddm_wks* is the old dynamic domain manager workstation name at the previous product version that now has the backup role.

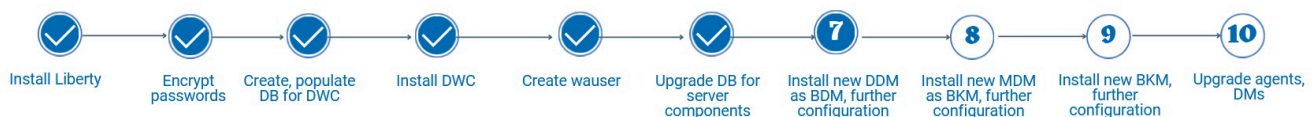
For more detailed information about switching a domain manager, see the complete procedure for switching a domain manager in *Administration Guide*.

What to do next

You can now proceed to install a new dynamic domain manager configured as a backup at the latest production version. as described in [Installing a new backup dynamic domain manager on page 345](#)

Installing a new backup dynamic domain manager

Procedure for installing the new backup dynamic domain manager



At this phase in the procedure, you have installed a fresh backup dynamic domain manager at the latest product version level and switched it to become the new dynamic domain manager. To complete the environment set up, you now need to install a new backup dynamic domain manager at the latest product version level by running the `serverinst` script.

The procedure to install the dynamic domain manager and backup dynamic domain manager is exactly the same, with the difference that it is performed on two different workstations and that each installation points to its local Open Liberty installation. HCL Workload Automation determines whether or not a dynamic domain manager is already present in the environment and proceeds to install a dynamic domain manager or backup dynamic domain manager accordingly.

The HCL Workload Automation administrator installs the dynamic domain manager as the backup. The following information is required:

Table 20. Required information

Command parameter	Information type	Provided in...
HCL Workload Automation information		
--wouser	HCL Workload Automation administrative user name	Creating the HCL Workload Automation administrative user on page 337
--wapassword	HCL Workload Automation administrative user password	
WebSphere Application Server Liberty Base information		
--wlpdir	WebSphere Application Server Liberty Base installation directory	Installing WebSphere Application Server Liberty on page 315
Security information		
--sslkeyfolder	location of converted certificates	Converting default certificates on page 313
--sslpassword	password of converted certificates	Converting default certificates on page 313

Before starting the installation, ensure the following steps have been completed:

1. [Converting default certificates on page 313](#). Because you are installing a dynamic domain manager, also copy locally the `jwt.crt` file created in the conversion procedure.
2. [Installing WebSphere Application Server Liberty on page 315](#) on the workstation where you plan to install the dynamic domain manager and on the workstation where you plan to install the backup dynamic domain manager.
3. [Encrypting passwords \(optional\) on page 317](#)

4. [Upgrading the database for the server components on page 338](#)
5. [Creating the HCL Workload Automation administrative user on page 337](#)

You can run the **serverinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. For more information about all **serverinst** parameters and default values, see [Server components installation - serverinst script on page 442](#).

A properties file named `serverinst.properties` is available if you do not want to type parameters in the command line. This is especially useful if you need to specify many parameters or if you want to reuse the file for several installations. The file is located in `image_location/TWS/interp_name`.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

If you need to modify any of the default values, edit the `serverinst.properties` file, but do not modify the `serverinst.template` file located in the same path.

To install the dynamic domain manager as a backup, perform the following steps:

1. Log in to the workstation where you plan to install.
2. Browse to the folder where the `serverinst` command is located:

On Windows operating systems

`image_location\TWS\interp_name`

On UNIX operating systems

`image_location/TWS/interp_name`

3. Start the installation specifying a typical set of parameters. In this case, default values are used for all remaining parameters:

On Windows operating systems

```
cscript serverinst.vbs --acceptlicense yes --rdbmstype db_type
--dbhostname db_hostname --dbport db_port --dbname db_name
--dbuser db_user --dbpassword db_password --wauser wa_user
--wapassword wa_password --componenttype DDM --domain domain_name
--master mdm_name --mdmbrokerhostname mdm_broker_host_name
--sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
--mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir\wlp
```

On UNIX operating systems

```
./serverinst.sh --acceptlicense yes --rdbmstype db_type
--dbhostname db_hostname --dbport db_port --dbname db_name
--dbuser db_user --dbpassword db_password --wauser wa_user
--wapassword wa_password --componenttype DDM --domain domain_name
--master mdm_name --mdmbrokerhostname mdm_broker_host_name
--sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
--mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir/wlp
```

4. Distribute the Symphony file to the new dynamic domain manager configured as backup:

- a. Ensure that the **optman cf** option is set to *all*.
- b. To distribute the Symphony file to the new dynamic domain manager configured as backup, run JnextPlan -for 0000 or wait until the end of the production plan.
- c. Restore the previous setting of the **optman cf** option, if you previously modified the value.

You have now successfully installed the backup dynamic domain manager at the new product version level.

For more information about all **serverinst** parameters and default values, see [Server components installation - serverinst script on page 442](#).

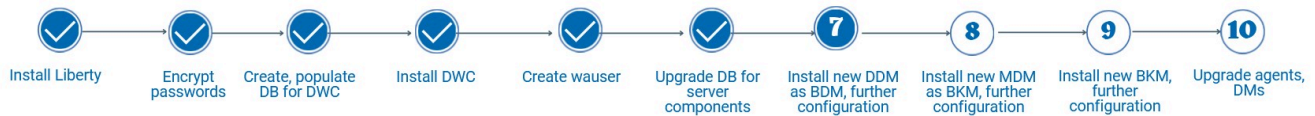
What to do next

You can now optionally proceed to [Switching back to the old dynamic domain manager \(optional\) on page 348](#).

Switching back to the old dynamic domain manager (optional)

Optionally switch back to the old dynamic domain manager

About this task



This step is optional. You can switch back to your old dynamic domain manager.

From the old dynamic domain manager, run the command:

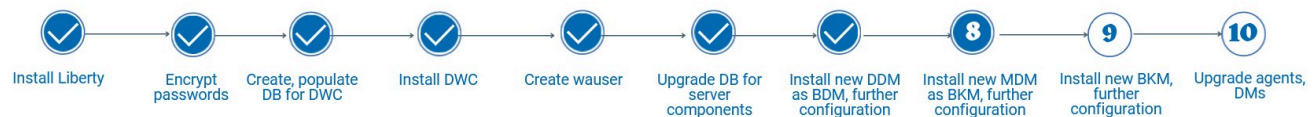
```
conman
switchmgr dyn_dom;old_mgr_cpu
```

where *dyn_dom* is the domain where the dynamic domain manager configured as backup is installed and the *old_mgr_cpu* is the old dynamic domain manager workstation name

Installing the new master domain manager configured as a backup

Install the new master domain manager configured as a backup and link it to your current network. Then switch it to become the new master domain manager.

Before you begin



About this task

Complete the steps listed below to install a fresh master domain manager configured as backup and then link it to your current network.

If the master domain manager has a version earlier than Version 95 Fix Pack 4, before performing the upgrade, run the **Flexera Analysis tool** to ensure that the LICENSE_JOBS_NUMBER parameter is equal to 0. For more information about the tool, see [Flexera Analysis tool](#).

The master domain manager configured as a backup points to your existing HCL Workload Automation database and then later becomes your new master domain manager.

During the master domain manager upgrade process, the license model to be applied to the environment is defined. The license model determines the criteria by which your license compliance is calculated. The following pricing models are supported: **byWorkstation**, **perServer**, **perJob**. The default value is **perServer**. To determine the current value of this global option, enter the following command: **optman show ln** or **optman show licenseType**. To modify the pricing model, use the **optman chg ln** or **optman chg licenseType** command. For more information about licensing, see License computation model the section about license computation model in *Administration Guide*.

1. [Converting default certificates on page 313](#), if you are using default certificates in your current environment. Use this procedure to convert the certificates from the JKS to the PEM format, then copy them to the workstations where you plan to install the server components (dynamic domain manager and its backups, master domain manager and its backups) and the Dynamic Workload Console.

Certificates are now required when installing or upgrading HCL Workload Automation. You can no longer install nor upgrade HCL Workload Automation without securing your environment with certificates. The required certificates are:

- ca.crt
- tls.key
- tls.crt

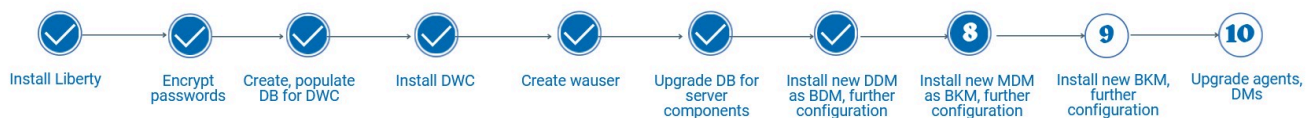
For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

2. [Installing the master domain manager as a backup master domain manager on page 349](#)
3. [Switching the master domain manager to the new backup master on page 355](#)
4. [Making the switch manager permanent on page 356](#)
5. [Customizing and submitting the optional FINAL job stream on page 358](#)

Installing the master domain manager as a backup master domain manager

A fresh installation for the master domain manager and the backup master domain manager

Before you begin



Before beginning the installation, ensure you have completed the following steps:

1. [Converting default certificates on page 313](#)
2. [Installing WebSphere Application Server Liberty on page 315](#)
3. [Encrypting passwords \(optional\) on page 317](#)
4. [Upgrading the database for the server components on page 338](#)
5. [Creating the HCL Workload Automation administrative user on page 337](#)

About this task

You install a master domain manager at the latest product version level configured as the new backup master domain manager by running the serverinst script. The installation process is able to detect the presence of an existing master domain manager and automatically configures this one as the backup master domain manager. The new backup master domain manager is configured to point to the existing database instance.

The HCL Workload Automation administrator installs the master domain manager as the backup. The following information is required:

Table 21. Required information

Command parameter	Information type	Provided in..
Database information		
--rdbmstype	database type	Upgrading the database for the server components on page 338
--dbhostname	database hostname	
--dbport	database port	
--dbname	database name	
--dbuser	database user name	
--dbpassword	database password	
HCL Workload Automation information		
--wauser	HCL Workload Automation administrative user name	Creating the HCL Workload Automation administrative user on page 337
--wapassword	HCL Workload Automation administrative user password	
WebSphere Application Server Liberty Base information		
--wlpdir	WebSphere Application Server Liberty Base installation directory	Installing WebSphere Application Server Liberty on page 315
HCL Workload Automation installation directory		

Table 21. Required information

(continued)

<code>--inst_dir</code>	installation directory	Current procedure
Security information		
<code>--sslkeysfolder</code>	location of converted certificates	Converting default certificates on page 313
<code>--sslpassword</code>	password of converted certificates	Converting default certificates on page 313

You can run the **serverinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. For more information about all **serverinst** parameters and default values, see [Server components installation - serverinst script on page 442](#).

A properties file named `serverinst.properties` is available if you do not want to type parameters in the command line. This is especially useful if you need to specify many parameters or if you want to reuse the file for several installations. The file is located in `image_location/TWS/interp_name`.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

If you need to modify any of the default values, edit the `serverinst.properties` file, but do not modify the `serverinst.template` file located in the same path.

To install the master domain manager as a backup, perform the following steps:

1. Log in to the workstation where you plan to install the master domain manager.
2. Browse to the folder where the `serverinst` command is located in `image_location/TWS/interp_name`.
3. Start the installation specifying a typical set of parameters. In this case, default values are used for all remaining parameters:

On Windows operating systems

```
cscript serverinst.vbs --acceptlicense yes --rdbmstype <db_type>
--dbhostname <db_hostname> --dbport <db_port> --dbname <db_name>
--dbuser <db_user> --dbpassword <db_password> --wauser <wa_user>
--wapassword <wa_password> --wlpdir <Liberty_installation_dir>\wlp
--sslkeysfolder <certificate_files_path> --sslpassword <keystore_truststore_password>
--inst_dir <installation_dir>
--licenseserverid <license_server_ID>
```

On UNIX operating systems

```
./serverinst.sh --acceptlicense yes --rdbmstype <db_type>
--dbhostname <db_hostname> --dbport <db_port> --dbname <db_name>
--dbuser <db_user> --dbpassword <db_password> --wauser <wa_user>
```

```
--wapassword <wa_password> --wlpdir <Liberty_installation_dir>/wlp
--sslkeysfolder <certificate_files_path> --sslpassword <keystore_truststore_password>
--inst_dir <installation_dir>
--licenseserverid <license_server_ID>
```

where

--acceptlicense

Specify **yes** to accept the product license.

--rdbmstype|-r *rdbms_type*

The database type. Supported databases are:

- **DB2**
- **ORACLE** This value applies to Oracle and Amazon RDS for Oracle
- **MSSQL** This value applies to MSSQL and MSSQL cloud-based databases.
- **POSTGRESQL**

This parameter is required and has no default value.

--dbhostname *db_hostname*

The host name or IP address of database server.

--dbport *db_port*

The port of the database server.

--dbname *db_name*

The name of the HCL Workload Automation database.

--dbuser *db_user*

The database user that has been granted access to the HCL Workload Automation tables on the database server.

--dbpassword *db_password*

The password for the user that has been granted access to the HCL Workload Automation tables on the database server. Special characters are not supported.

--wauser *user_name*

The user for which you are installing HCL Workload Automation.

--wapassword *wauser_password*

The password of the user for which you are installing HCL Workload Automation.

On Windows operating systems

Supported characters for the password are alphanumeric, dash (-), underscore (_), characters, and ()|?*~+.@!^

On UNIX operating systems

Supported characters for the password are any alphanumeric, dash (-), underscore (_), characters, and ()|?*~+.

--wlpdir

The path where Open Liberty is installed.

--licenseserverid

The ID of the license server which processes license usage information. This parameter is required. For more information about enabling your product license, see [Enabling product license management on page 52](#).

--sslkeysfolder *keystore_truststore_folder*

The name and path of the folder containing certificates in PEM format. The installation program automatically processes the keystore and truststore files using the password you specify with the **--sslpassword** parameter. The folder must contain the following files:

- **ca.crt**

The Certificate Authority (CA) public certificate. Note that if certificates being installed are part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then this file must contain the Root CA certificate only. Any Intermediate CA certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.



Note: From V10.2.3, if certificates being installed are part of a chain, the `ca.crt` can contain also the intermediate CAs. In this case, it must begin with one or more intermediate CA certificates and end with the Root ca.

- **tls.key**

The private key of the end user certificate for the instance to be installed.

- **tls.crt**

The public part of the previous key, that is the end user certificate.

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

You can optionally create a subfolder to contain one or more `*.crt` files to be added to the server truststore as trusted CA, whose name must be `additionalCAs`. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or DB2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**. Note that if the end user certificate being installed in the instance is part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then the Intermediate CAs certificates must be stored in the

`additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

For further information about how to generate custom certificates, see the topic about managing certificates using Certman in *HCL Workload Automation: Planning and Installation*.

--sslpassword *ssl_password*

The password for the certificates.

For more information, see [sslkeysfolder on page 450](#).

You can optionally encrypt the password using the secure script. For more information, see [Optional password encryption - secure script on page 427](#).

--inst_dir *installation_dir*

The directory of the HCL Workload Automation installation.

--licenseserverid *license_server_ID*

The ID of the license server which processes license usage information. This parameter is required. Instructions about how to obtain the ID of the license server which processes license usage information are provided with the mail confirming your license. For more information, see the section about License computation model in *Administration Guide* and Enabling product license management in *HCL Workload Automation: Planning and Installation*.



Note: The values for the following parameters must match the values you provided when creating and populating the database:

- **--rdbmstype**
- **--dbhostname**
- **--dbport**
- **--dbname**
- **--dbuser**
- **--dbpassword**



Note: Before starting the deployment of a new master domain manager or backup master domain manager on an already used database, be sure that no failed plan creation/extension has been performed. If a failed plan creation or extension has been performed, resolve the failure before attempting the new deployment or unlock the database by running the `planman unlock db` command.

4. If you are installing a backup master domain manager, it is crucial to use the same encryption keys as those on the master domain manager, to ensure it can correctly decrypt encrypted files, such as the Symphony file. To achieve this, perform the following steps:

- a. Backup the files located in the `TWA_DATA_DIR\ssl\aes` folder on the backup master domain manager.
 - b. Copy the files from the `TWA_DATA_DIR\ssl\aes` folder on the master domain manager to the `TWA_DATA_DIR\ssl\aes` folder on the backup master domain manager.
5. To verify that the installation completed successfully, browse to the directory where you installed the master domain manager and type the following commands:

On UNIX operating systems

```
./twc_env.sh
```

On Windows operating systems

```
twc_env.cmd
```

```
optman ls
```

This command lists the HCL Workload Automation configurations settings and confirms that HCL Workload Automation installed correctly.

You can also optionally run `JnextPlan -for 0000` to extend by 0 hours and 0 minutes the production plan and add into the production plan (Symphony) the newly-created workstation, or wait for the FINAL job stream to complete, then run `composer list cpu=server_workstation_name` to ensure the agents have registered. You can also run a test job to ensure everything is working correctly.

Results

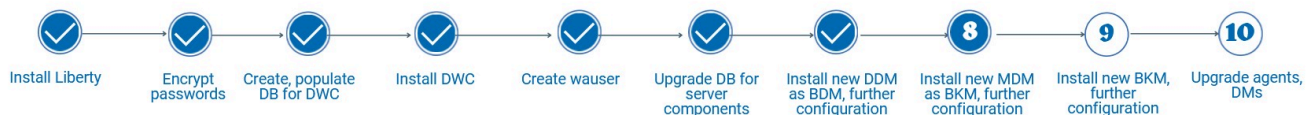
You have now successfully installed the master domain manager as the backup master domain manager.

What to do next

You can now proceed to [Switching the master domain manager to the new backup master on page 355](#).

Switching the master domain manager to the new backup master

About this task



To switch the back-level master domain manager to the new backup master domain manager, complete the following procedure:

1. Start WebSphere Application Server Liberty Base on the new backup master domain manager by running the `startAppServer` script found in the following path:

```
<TWA_HOME>/appservertools/startAppServer.sh
```

2. Before you switch your master domain manager to the new backup master domain manager, you must stop the dynamic workload broker server on the current back-level master domain manager:

On Windows™ operating systems

Use `wastool stopBrokerApplication.bat`

On UNIX® operating systems

Use `wastool stopBrokerApplication.sh`

3. Switch to your new backup master domain manager, which now becomes your current active master domain manager, by issuing the following command from either the Dynamic Workload Console or the **command line** of your old master domain manager:

From the Dynamic Workload Console

In the navigation tree, click **Monitoring and Reporting > Monitor Workload >** select the engine and the object type Workstation, click run and, in the table of results, select backup master domain manager workstation name, click **More Actions**, and select **Become Master Domain Manager**.

From the command line of the old master domain manager

Issue the following command:

```
conman "switchmgr masterdm;new_mgr_cpu"
```

where *new_mgr_cpu* is the backup master domain manager workstation name.

4. Switch the event processor from the old master domain manager to the backup master domain manager, by running the following command from either the Dynamic Workload Console or the **command line** of your old master domain manager:

From the Dynamic Workload Console

In the navigation tree, click **Monitoring and Reporting > Monitor Workload >** select the engine and the object type Workstation, click run and, in the table of results, select backup master domain manager workstation name, click **More Actions**, and select **Become Event Processor**.

From the command line of the old master domain manager

Issue the following command:

```
conman "switcheventprocessor new_mgr_cpu"
```

where *new_mgr_cpu* is the backup master domain manager workstation name.

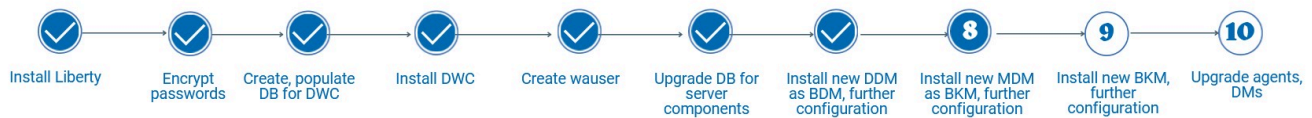
Results

Once you have switched the master domain manager to the new backup master, you can make this switch permanent. For details, see [Making the switch manager permanent on page 356](#).

For more detailed information about switching the master domain manager, see the related topic in the *Administration Guide*

Making the switch manager permanent

About this task



In the procedure [Switching the master domain manager to the new backup master on page 355](#), you switched your master domain manager to the new backup master domain manager promoting your new version backup master domain manager to the role of master domain manager.

To make this configuration fully operational and persistent through **JnextPlan**, you must complete the following procedure:

On the new master domain manager, referred to as *new_mgr_cpu*, perform the following steps:

1. Edit the *localopts* file and modify the following entry as shown:

```
DEFAULTWS=new_mgr_cpu
```

where *new_mgr_cpu* is the workstation name of the new master domain manager. For more information about *localopts* file, see the section about setting local options in *Administration Guide*.

2. Change the workstation definition of the back-level master by running:

```
composer modify cpu=old_mgr_cpu
```

and in the definition substitute *type=manager* with *type=fta*

3. Change the workstation definition of the new master by running:

```
composer modify cpu=new_mgr_cpu
```

and in the definition substitute *type=fta* with *type=manager*.

4. Ensure that the **optman cf** option is set to *all*.
5. Rebuild the plan to activate the changes to the database:

```
JnextPlan -for 0000
```

6. Switch the event processor to the new master domain manager by running the following command:

```
switcheventprocessor new_mgr_cpu
```

7. Restore the previous setting of the **optman cf** option, if necessary.
8. Edit the *TWA_DATA_DIR/mozart/globalopts* file and modify the **master=old_mgr_cpu** entry as shown:

```
master=new_mgr_cpu
```

where *new_mgr_cpu* is the workstation name of the new master. For more information about **optman**, see the section about setting global options in *Administration Guide*.

In this way the reports *reptr-pre* and *reptr-post* can run when you run **JnextPlan**.

What to do next

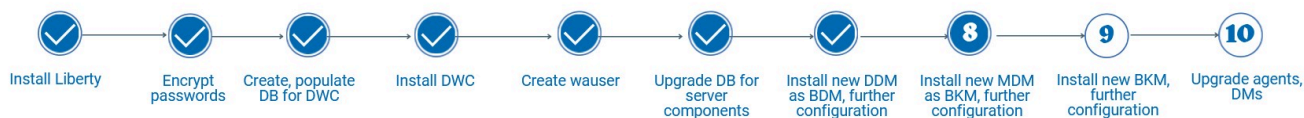
Once you have made the switch manager permanent, you must run the FINAL job stream on the new master domain manager.

You can now proceed to [Customizing and submitting the optional FINAL job stream on page 358](#).

Customizing and submitting the optional FINAL job stream

Merge the functions of your current FINAL and FINALPOSTREPORTS job streams with the syntax of your new FINAL and FINALPOSTREPORTS job streams.

About this task



The upgrade process writes the latest FINAL and FINALPOSTREPORTS definitions for the current release in the following file: `<TWA_HOME>/TWS/config/Sfinal`, where `<TWA_HOME>` is the HCL Workload Automation installation directory. To use these latest definitions, you must merge the functions of your current FINAL and FINALPOSTREPORTS job streams with the syntax of your new FINAL and FINALPOSTREPORTS job streams.



Important: The definitions of the FINAL and FINALPOSTREPORTS job streams in `<TWA_HOME>/TWS/config/Sfinal` are defined on an extended agent that might not be defined in the new environment. If you are planning to use the old definitions to replace the new ones using the `composer replace` command, you must either change the workstation on which the jobs are defined to an existing one, or you must create a new extended agent where the jobs inside the *Sfinal* are defined.

Complete the following procedure:

1. Depending on your situation, edit your current final job streams and customize the new final job streams as follows:

If you had customized job streams called FINAL and FINALPOSTREPORTS in your database:

- a. Extract the definitions from the current FINAL and FINALPOSTREPORTS job streams file by using `composer`.
- b. Use a text editor to edit your customized FINAL and FINALPOSTREPORTS job streams.
- c. Merge the job streams with file `<TWA_HOME>/TWS/config/Sfinal` so that the new FINAL and FINALPOSTREPORTS job streams have the same customization as your customized final job streams plus the new required attributes provided by the new FINAL and FINALPOSTREPORTS job streams.
- d. Save your new FINAL and FINALPOSTREPORTS job streams by using `composer`.

If you had customized final job streams called something other than FINAL and FINALPOSTREPORTS in your database:

- a. Extract the definitions from your customized final job stream files by using `composer`.
- b. Use a text editor to edit your customized final job stream files.
- c. Merge the job streams with file `<TWA_HOME>/TWS/config/Sfinal` so that the new FINAL and FINALPOSTREPORTS job streams have the same customization as your customized final job

streams plus the new required attributes provided by the new FINAL and FINALPOSTREPORTS job streams.

- d. Save these new final job streams so that they have the same names as your current customized final job streams by running the command `composer replace`.

If you had final job streams called something other than FINAL and FINALPOSTREPORTS in your database, but they are not customized:

- a. Make a copy of file `<TWA_HOME>/TWS/config/Sfinal`.
- b. Edit this copy and rename the FINAL and FINALPOSTREPORTS parameters with the actual names.
- c. Run the command `composer replace`.

If you had final job streams called FINAL and FINALPOSTREPORTS in your database, but they are not customized:

Run the command `composer replace <TWA_HOME>/TWS/config/Sfinal`.

If you had final job streams called FINAL and FINALPOSTREPORTS but they are in DRAFT in your database:

Run the command `composer replace` and, after the upgrade, change these job streams into the DRAFT status again.

2. After you customized the new final job streams, you must delete your current final job stream instances (`conman cancel sched command`) and submit the new final job stream instances (`conman sbs sched command`).

During the upgrade, JnextPlan is overwritten even if you customized it. The existing JnextPlan is backed up and renamed to:

On Windows™ operating systems:

JnextPlan.cmd.bk

On UNIX™ and Linux™ operating systems:

JnextPlan.bk

Installing a new backup master domain manager

Upgrading your old master domain manager, which is now your current backup master domain manager to the latest product version level.

About this task



Now that you have a new master domain manager installed at the latest product version level, you can upgrade your old, previous version master domain manager, which is currently your backup master domain manager, to the latest product

version to become the new backup master domain manager. You do this by installing a new backup master domain manager. Ensure you specify the same user as the one specified for the master domain manager.

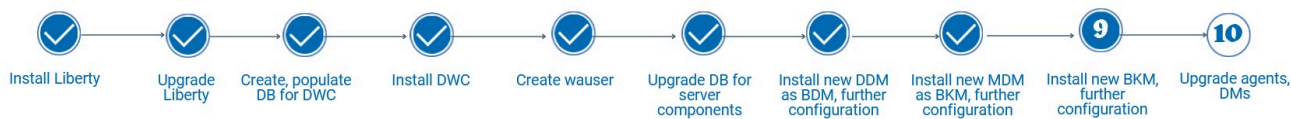


Note: If you want to minimize the number of workstations required, you can install the new backup master domain manager on the same workstation where your old master domain manager was running. Ensure you stop any running processes related to the previous product version before installing the new backup master domain manager..

Installing a new backup master domain manager

Installing the new backup master domain manager

Before you begin



Before beginning the installation, ensure you have converted the certificates, as described in [Converting default certificates on page 313](#).

About this task

You can perform a typical installation, as described in the following scenario, or you can customize the installation parameters, as described in [FAQ - master domain manager and backup master domain manager customizations on page 107](#).

For more information about all **serverinst** parameters and default values, see [Server components installation - serverinst script on page 442](#).

The procedure to install the backup master domain manager is exactly the same as installing a master domain manager. The backup master domain manager is installed on a workstation different from the master domain manager and points to its local WebSphere Application Server Liberty Base installation. HCL Workload Automation detects the presence of an existing master domain manager in the environment and proceeds to install a backup master domain manager.

The HCL Workload Automation administrator installs the master domain manager. The following information is required:

Table 22. Required information

Command parameter	Information type	Provided in..
HCL Workload Automation information		
--wauser	HCL Workload Automation administrative user name	Creating the HCL Workload Automation administrative user on page 337

Table 22. Required information

(continued)

--wapassword	HCL Workload Automation administrative user password	
WebSphere Application Server Liberty Base information		
--wlpdir	WebSphere Application Server Liberty Base installation directory	Installing WebSphere Application Server Liberty on page 315
Security information		
--sslkeysfolder	location of converted certificates	Converting default certificates on page 313
--sslpassword	password of converted certificates	Converting default certificates on page 313

You can run the **serverinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. For more information about all **serverinst** parameters and default values, see [Server components installation - serverinst script on page 442](#).

A properties file named `serverinst.properties` is available if you do not want to type parameters in the command line. This is especially useful if you need to specify many parameters or if you want to reuse the file for several installations. The file is located in `image_location/TWS/interp_name`.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

If you need to modify any of the default values, edit the `serverinst.properties` file, but do not modify the `serverinst.template` file located in the same path.

To install the backup master domain manager, perform the following steps:

1. Log in to the workstation where you plan to install.
2. Browse to the folder where the `serverinst` command is located in `image_location/TWS/interp_name`.
3. Start the installation specifying a typical set of parameters. In this case, default values are used for all remaining parameters:

On Windows operating systems

```
cscript serverinst.vbs --acceptlicense yes --rdbmstype <db_type>
--dbhostname <db_hostname> --dbport <db_port> --dbname <db_name>
--dbuser <db_user> --dbpassword <db_password> --wauser <wa_user>
--wapassword <wa_password> --wlpdir <Liberty_installation_dir>\wlp
--sslkeysfolder <certificate_files_path> --sslpassword <keystore_truststore_password>
```

On UNIX operating systems

```
serverinst.sh --acceptlicense yes --rdbmstype <db_type>
--dbhostname <db_hostname> --dbport <db_port> --dbname <db_name>
--dbuser <db_user> --dbpassword <db_password> --wauser <wa_user>
--wapassword <wa_password> --wlpdir <Liberty_installation_dir>/wlp
--sslkeysfolder <certificate_files_path> --sslpassword
<keystore_truststore_password>
```

where

acceptlicense

Specify **yes** to accept the product license.

rdbmstype|-r rdbms_type

The database type. Supported databases are:

- DB2
- ORACLE
- MSSQL

This parameter is optional. The default value is **db2**.

dbhostname db_hostname

The host name or IP address of database server.

dbport db_port

The port of the database server.

dbname db_name

The name of the HCL Workload Automation database.

dbuser db_user

The user that has been granted access to the HCL Workload Automation tables on the database server.

dbpassword db_password

The password for the user that has been granted access to the HCL Workload Automation tables on the database server. Special characters are not supported.

wauser user_name

The user for which you are installing HCL Workload Automation.

wapassword wauser_password

The password of the user for which you are installing HCL Workload Automation.

On Windows operating systems

Supported characters for the password are alphanumeric, dash (-), underscore (_), characters, and ()|?*~+.@!^

On UNIX operating systems

Supported characters for the password are any alphanumeric, dash (-), underscore (_), characters, and ()|?=*~+.

wlpdir

The path where WebSphere Application Server Liberty Base is installed.

--sslkeyfolder *keystore_truststore_folder*

The name and path of the folder containing certificates in PEM format. The installation program automatically processes the keystore and truststore files using the password you specify with the --**sslpassword** parameter. The folder must contain the following files:

- **ca.crt**

The Certificate Authority (CA) public certificate. Note that if certificates being installed are part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then this file must contain the Root CA certificate only. Any Intermediate CA certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.



Note: From V10.2.3, if certificates being installed are part of a chain, the `ca.crt` can contain also the intermediate CAs. In this case, it must begin with one or more intermediate CA certificates and end with the Root ca.

- **tls.key**

The private key of the end user certificate for the instance to be installed.

- **tls.crt**

The public part of the previous key, that is the end user certificate.

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

You can optionally create a subfolder to contain one or more *.`crt` files to be added to the server truststore as trusted CA, whose name must be `additionalCAs`. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or DB2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**. Note that if the end user certificate being installed in the instance is part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then the Intermediate CAs certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

For further information about how to generate custom certificates, see the topic about managing certificates using Certman in *HCL Workload Automation: Planning and Installation*.

--sslpassword *ssl_password*

The password for the certificates.

For more information, see [sslkeysfolder on page 450](#).

You can optionally encrypt the password using the secure script. For more information, see [Optional password encryption - secure script on page 427](#).

4. To verify that the installation completed successfully, browse to the directory where you installed the backup master domain manager and type the following commands:

```
. ./twc_env.sh
```

```
optman ls
```

This command lists the HCL Workload Automation configurations settings and confirms that HCL Workload Automation installed correctly.

Results

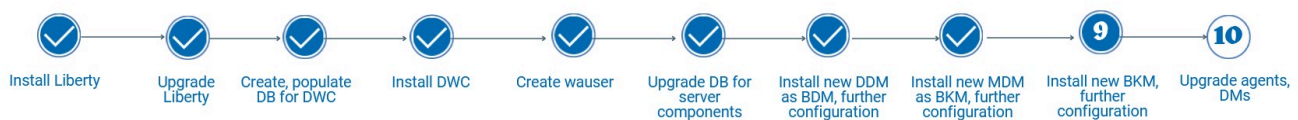
You have now successfully installed the backup master domain manager and it is inserted in the next production plan. To have the backup domain manager added immediately to the production plan, run

```
jnextPlan -for 0000
```

What to do next

You can now proceed to [Ensuring communication in your environment on page 364](#).

Ensuring communication in your environment



Security is enabled by default starting from version 10.1, but is usually not configured in most back-level environments. If security is not configured in your current environment, perform the following steps to ensure all HCL Workload Automation components can communicate correctly:

Most 9.4 environments are not configured with SSL, which is enabled by default starting from version 10.1. To ensure communication between all components, perform the following steps:

1. On the backup master domain manager at version 10.2.5, stop WebSphere Application Server Liberty, as described in the topic about starting and stopping the application server in *Administration Guide*.
2. Browse to the following paths:

on Windows operating systems

```
TWS\broker\config
```

on UNIX operating systems

```
TWS/broker/config
```

3. Set the **Broker.Workstation.PortSSL** property to `false` in the `BrokerWorkstation.properties` file.
4. Start WebSphere Application Server Liberty on the backup master domain manager at version 10.2.5, as described in the topic about starting and stopping the application server in *Administration Guide*.
5. Run the following commands on the back-level master domain manager:

- a. `optman chg cf = ALL`

This command changes the **enCarryForward** option so that all incomplete job streams are carried forward.

- b. `JnextPlan -for 0000 -noremove`

This command extends the production plan without removing successfully completed job stream instances.

- c. `optman chg cf = <original value>`

This command returns the **enCarryForward** option to its original value.

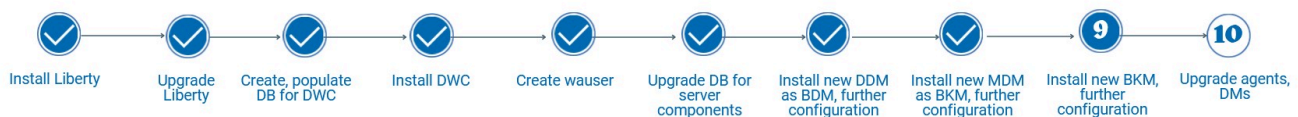
The new backup master domain manager can now communicate with the back-level network.

If you want to switch the new backup master domain manager to master, stop the broker on the back-level master domain manager, and switch it to master domain manager.

What to do next: You can now optionally proceed to [Uninstalling the back-level backup master domain manager on page 365](#).

Uninstalling the back-level backup master domain manager

Procedure to uninstall the back-level backup master domain manager

About this task

1. Ensure that the user running the process has the following authorization requirements:

Windows™ operating systems

If you set the Windows User Account Control (UAC), your login account must be a member of the Windows™ **Administrators** group or domain administrators with the right, **Act as Part of the Operating System**.

If you set the Windows User Account Control (UAC) on the workstation, you must run the installation as **administrator**.

UNIX™ and Linux™ operating systems

If the component was installed with root privileges, **root** access is required. If you performed a **no-root installation**, specify the same user used for installing the component.

2. Ensure that all HCL Workload Automation processes, services and the WebSphere Application Server Liberty process are stopped, and that there are no active or pending jobs. For information about stopping the processes and services see the topic about starting and stopping processes on a workstation in the *User's Guide and Reference*.

To uninstall a backup master domain manager, perform the following steps:

1. To uninstall the backup master domain manager, you must first remove it from the plan. Set the workstation running the backup master domain manager to `ignore`, using either the `composer mod cpu workstation_name>` command or from the Dynamic Workload Console.
2. Run JnextPlan to generate the new production plan so that the backup master domain manager is removed from the plan.
3. Run the uninstall script.

- a. Change directory using the following command:

```
cd TWA_home>/TWS/tws_tools
```

- b. Run the uninstallation process by running the script as follows:

Windows™ operating systems

```
cscript uninstall.vbs --prompt no --wauser user_name>
```

UNIX™ and Linux™ operating systems

```
./uninstall.sh --prompt no --wauser user_name
```

where, `user_name>` represents the user for which you want to uninstall the backup master domain manager. The procedure runs without prompting the user to confirm the uninstallation.

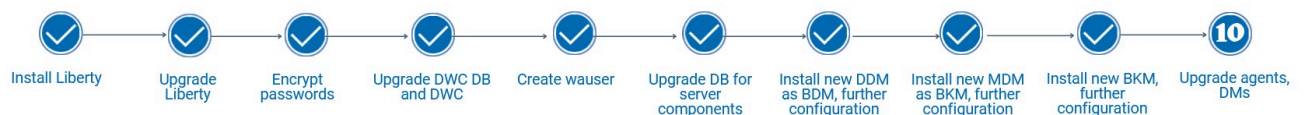
4. Run JnextPlan to update the plan with the changes.

What to do next

You can now proceed to [Upgrading agents and domain managers on page 366](#).

Upgrading agents and domain managers

There are several methods you can choose from to upgrade your domain managers and



agents.

The agent upgrade can be performed with minimal impact to scheduling activities. The agents are stopped for the shortest time necessary to perform the maintenance. Any active agent command-line interfaces and processes, such as conman, composer, netman, mailman, and batchman, to name a few, continue running. Any jobs already running when the upgrade process begins, continue to run as planned, however, no new jobs begin execution during this time. Once the upgrade is complete, the agent is restarted and quickly reconnects with its jobs. Any jobs that were actively running before the upgrade that have not yet completed, continue to run, and any jobs that successfully finished running during the upgrade procedure report a successful job status. An automatic backup and restore feature is in place in case of failure.

Because domain managers are agents, they are upgraded using the procedures described in this section.

If you choose to upgrade your environment top-down, then the agents get upgraded progressively after you have upgraded the master domain manager and its backup. This means that new features and enhancements are not available on all of your agents at the same time. If, instead, you choose to upgrade your environment bottom-up, then the agents are upgraded first, and new features and enhancements become available after the master domain manager and its backup have been upgraded.



Important: After upgrading your fault-tolerant agents, it might be necessary to 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. See the topic about updating the security file in the *Administration Guide* for more information.

You can choose to upgrade your agents using any of the following methods:

twinsinst script

A single line command that checks if processes or a command line is running before it starts. It saves disk space and RAM because it is not Java-based. See [Upgrade procedure on page 290](#) and [Upgrading agents on IBM i systems on page 295](#)

Centralized agent update

Upgrade or update multiple fault-tolerant agent and dynamic agent instances at the same time. Download the fix pack installation package, or the elmage upgrade package to the master domain manager and then either run the installation on multiple agent instances or schedule the installation by creating and submitting a job to run. This upgrade method is not supported on z-centric agent instances. See [Centralized agent update on page 300](#).

For a list of supported operating systems and requirements, see the System Requirements Document at [HCL Workload Automation Detailed System Requirements](#).

When the upgrade procedure has completed successfully, the backup instance is deleted.



Note: 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 following path:



On Windows operating systems

<TWA_home>\TWS

On UNIX operating systems

<TWA_DATA_DIR>

When upgrading dynamic agents featuring both a local and a remote gateway, ensure you either upgrade the agent first and then the gateway or upgrade both at the same time.

Related information

[Upgrading the database for the server components on page 338](#)

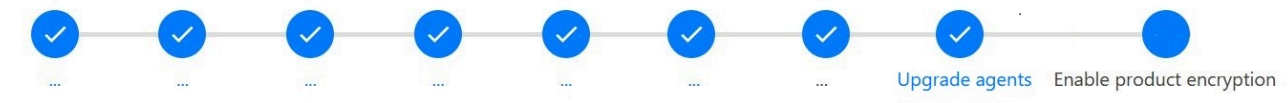
[Enabling product encryption after upgrading on page 368](#)

Enabling product encryption after upgrading

Enabling product encryption after upgrading from a version earlier than 10.1.

About this task

If you are upgrading from a version earlier than version 10.1, you can optionally enable encryption for key product files by performing the following steps on the master domain manager and on each agent in the environment:



1. Generate a new key by running the following keytool command:

```
./keytool -genseckey -alias new_alias_name -keyalg AES -keysize 256
-storepass encrypt_keystore_pwd_in_clear -storetype PKCS12 -keystore encrypt_keystore_file
```

2. Create the stash file containing a password encoded in base64. You can store the file in a path of your choice.
3. Add the following keys in the `localopts` file:

encrypt keystore file *file_name*

The path to the keystore PKCS12 file, containing the AES-256 or AES-128 key. The keystore is created automatically at installation time and the related path is inserted in this parameter. If you want to use a different keystore, you can create it and add the path in this option.

encrypt keystore pwd *password*

The path to the keystore stash file.

encrypt label

The label you assign to the new key in the keystore. This property is case insensitive.

Consider the following example of the modifications to the `localopts` file:

```
encrypt keystore file ="/opt/wa/TWA/TWS/ssl/key.p12"
encrypt keystore pwd ="/opt/wa/TWA/TWS/ssl/key.sth"
encrypt label ="myalias"
```

where

encrypt keystore file

corresponds to the **-keystore** *encrypt_keystore_file* parameter in the command provided in step 1.

encrypt keystore pwd

corresponds to the path of the stash file created in step 2.

encrypt label

corresponds to the **-alias** *new_alias_name* parameter in the command provided in step 1.

Results

The current Symphony plan keeps using the previous key. To apply the new setting to the Symphony plan, run a JnextPlan command. The message boxes are encrypted immediately and the `useropts` file is encrypted as soon as you save the `localopts` file and launch a CLI command. Key product files are now encrypted with the new key.

Enabling API Key authentication after upgrading

Enabling API Key authentication after upgrading from v 10.x.x or v 9.5.x to 10.2.x.

About this task

In previous versions of the product, both in fresh and upgrade installation, it was not necessary to add the server public certificate to its truststore. With the new API Key feature, which is implemented in version 10.1 Fix Pack 1 and later, the generated JWT is signed with the server private key. When the JWT is received by the server to authenticate a user, the public key associated with the private key used for signing is not present in the truststore and cannot be used. As a result, the authentication of that user is blocked.

To solve the problem, in fresh installations the server public key is automatically added to its truststore.

When you are upgrading from v 10.x.x or v 9.5.x to 10.2.x, run the following commands on the master domain manager:

1. Export the certificates, as follows:

```
keytool -exportcert -keystore
$WA_DATADIR/usr/servers/engineServer/resources/security/TWSServerKeyFile.p12
-storepass password -storetype pkcs12 -file /tmp/tls.crt -alias server -noprompt
```

2. Import the certificates, as follows:

```
keytool -importcert -keystore
$WA_DATADIR/usr/servers/engineServer/resources/security/TWSServerTrustFile.p12
-storepass password -storetype pkcs12 -file /tmp/tls.crt -alias mpjwtkey -noprompt
```

3. Edit the value of the **mp.jwt.trust.key** variable from the **twstrustkey** to **mpjwtkey** in the `jwt_variables.xml` file located inside the WebSphere Application Server Liberty Base `overrides` folder. For more information about templates, see the topic about configuring HCL Workload Automation using templates in *Administration Guide*.

If you do not remember what the public certificate alias is called, run the following command to retrieve the list of certificates within the keystore:

```
keytool -list -keystore $WA_DATADIR/usr/servers/engineServer/resources/security/TWSServerKeyFile.p12
-storepass password -storetype pkcs12
```

Upgrading when there are corrupt registry files

If you have tried to upgrade a stand-alone, fault-tolerant agent (an agent that is not shared with other components or does not have the connector feature) and received an error message that states that an instance of HCL Workload Automation cannot be found, this can be caused by a corrupt registry file. It is possible to upgrade a stand-alone, fault-tolerant agent that has corrupt registry files without having to reinstall the product. HCL Workload Automation has a recovery option you can run to re-create the necessary files. You can also use this option when upgrading nodes in clusters, where the node on which you want to perform the upgrade is not available or is in an inconsistent state. The recovery option re-creates the registry files and the Software Distribution information without having to reinstall the complete product.

You can run the recovery option using the **twsinst** script.

Re-creating registry files using twsinst

To re-create the registry files while upgrading an agent by using the **twsinst** script, from the directory that contains the HCL Workload Automation agent image, run **twsinst** using the synopsis described below.

Synopsis:

On Windows™ operating systems:

Show command usage and version

```
twsinst -u | -v
```

Upgrade an instance

```
twsinst -update -uname user_name -password password
..-acceptlicense yes|no
[-domain user_domain]
[-recovInstReg true]
[-inst_dir install_dir]
```

Example

```
cscript twsinst -update -uname twsuser -password twspassword
-acceptlicense yes -inst_dir "C:\Program Files\IBM\TWA"
-recovInstReg true
```

On UNIX™ and Linux™ operating systems

Show command usage and version

```
./twsinst -u | -v
```

Upgrade an instance

```
./twsinst -update -uname user_name
..-acceptlicense yes|no
```

```
.. [-inst_dir install_dir
.. [-recovInstReg true]]
```

Example

```
./twsinst -update -uname twsuser -inst_dir /opt/IBM/TWA
-acceptlicense yes -recovInstReg true
```

For information about the **twsinst** parameters, see [Upgrade procedure on page 290](#).

Upgrading in a mixed-version environment when using default certificates

Upgrading in a mixed-version environment when using default certificates

About this task

If your environment contains components, such as agents, Dynamic Workload Console, dynamic domain managers, and so on, at various version levels and you use default certificates, ensure certificates across the environment are consistent.

For example, you might need to install an agent at version 10.2.x, and connect it to a back-level master domain manager.

If you are using default certificates, you need to convert them to the new format and make them available to all components before you start the upgrade, as described in the following steps:

1. Set the HCL Workload Automation environment, as described in [Setting the environment variables on page 206](#).
2. To ensure the keytool and openssl commands start correctly on all operating systems, browse to the folder where the keytool and openssl commands are located and launch the commands as follows:

```
cd <TWS_DIR>/JavaExt/jre/jre/bin

./keytool -importkeystore -srckeystore TWSServerKeyFile.jks -destkeystore
<path_of_extracted_certs>/server.p12 -deststoretype pkcs12

cd <TWS_DIR>/tmpOpenSSL64/1.1/bin/openssl

./openssl pkcs12 -in <path_of_extracted_certs>/server.p12 -out
<path_of_extracted_certs>/tls.tot
```

The location of the `TWSServerKeyFile.jks` varies depending on the HCL Workload Automation version you have currently installed, as follows:

versions 9.5 and later

```
TWA_DATA_DIR/usr/servers/engineServer/resources/security
```

versions 9.4 and earlier

```
TWA_home/WAS/TWSPProfile/etc
```

3. Open the `tls.tot` file with any text editor.
4. From the `tls.tot` file, copy the private key to a new file named `tls.key`.

The `tls.key` file must be structured as follows:

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
<private_key>
-----END ENCRYPTED PRIVATE KEY-----
```



Note: Insert a carriage return after each key, so that an empty line is inserted after each key.

- From the `tls.tot` file, copy the public key to a new file named `tls.crt`.

The `tls.crt` file must be structured as follows:

```
-----BEGIN CERTIFICATE-----
<public_key>
-----END CERTIFICATE-----
```



Note: Insert a carriage return after each key, so that an empty line is inserted after each key.

- Copy the contents of the `tls.crt` file into a new file named `ca.crt`. If you want to upgrade a dynamic domain manager, also copy the contents of the `tls.crt` file into another new file named `jwt.crt`.
- Create a file named `tls.sth` containing the passphrase you have specified for creating the `.p12` certificate in step 2 on page 371, encoded in `base64` format. To create the `tls.sth` file, use the following command:

```
./secure -password your_password -base64 e -out
<path_of_extracted_certs>/tls.sth
```

If you are using a version earlier than 10.x, you can find the `secure` script in the installation package of the 10.2.5 version you are upgrading to. You can launch the script from one of the following paths:

master domain manager and agent

```
<10.2.5_extracted_image_dir>/TWS/<interp>/Tivoli_LWA-<interp>/TWS/bin
```

Dynamic Workload Console

```
<10.2.5_extracted_image_dir>/DWC/<interp>/bin
```

where

<interp>

is the operating system you are installing on

As an alternative, you can use the following command on UNIX workstations:

```
echo -n "passwordToEncode" | base64 >> tls.sth
```

- Browse to the GSKit folder and extract the client certificates from the `TWA_DATA_DIR/ssl/GSKit` folder by running the following commands, depending on the HCL Workload Automation version you have currently installed:

```
cd <TWS_DIR>/tmpGSKit64/8/bin
```

versions 9.5 and later

```
./gsk8capicmd_64 -cert -extract -db <TWA_DATA_DIR>/ssl/GSKit/TWSCClientKeyStore.kdb
-stashed -label client -target <path_of_extracted_certs>/client.crt
```

versions 9.4 and earlier

```
./gsk8capicmd_64 -cert -extract -db <TWS_DIR>/ssl/GSKit/TWSCClientKeyStore.kdb
-stashed -label client -target <path_of_extracted_certs>/client.crt
```


9. Create a folder named `additionalCAs` in the folder where you extracted the certificates and move the `client.crt` file created in [step 8 on page 372](#) to the `additionalCAs` folder.
10. Insert the `client.crt` in the `additionalCAs` folder when providing the certificates to the installation script with the **`sslkeysfolder`** parameter.
11. Assign the correct permissions (755) and ownerships to extracted certificates, as follows:

```
chmod -R 755 <path_of_extracted_certs>
```

Results

You have now converted the certificates to the required PEM format.

What to do next

You can now use the new default certificates for installing or upgrading HCL Workload Automation components, as follows:

If your master domain manager is at least at 10.1 FP1 level

you can copy the certificates you converted with the above procedure to the `/depot` folder on the master domain manager and install or upgrade dynamic agents and fault-tolerant agents specifying the **`wauser`** and **`wapassword`** parameters. For all remaining components, copy the certificates locally and launch the installation or upgrade specifying the **`sslkeysfolder`** and **`sslpassword`** parameters.

If your master domain manager is at a version earlier than 10.1 FP1 level

copy the certificates you converted with the above procedure locally on all components and launch the installation specifying the **`sslkeysfolder`** and **`sslpassword`** parameters.

For more information about all installation and upgrade parameters, see the `serverinst`, `dwcinst`, and `twcinst` scripts in [Reference on page 427](#).

Chapter 20. Updating containers

Updating the container configuration parameters.

To change the container configuration parameters or to obtain the latest version of a container, an update is required.

Complete the following procedure to update a Docker container:

1. Contact your HCL sales representative for the login details required to access the HCL Entitled Registry
2. Run the following command to log into the HCL Entitled Registry:

```
docker login -u <your_username> -p <your_entitled_key> hclcr.io
```

The console image is named `hclcr.io/wa/hcl-workload-automation-console:<release_name>`

3. Run the following command to log into the HCL Entitled Registry:

```
docker login -u <your_username> -p <your_entitled_key> hclcr.io
```

4. Manually update the compose file by modifying the *image* name if docker-compose does not reference the version to which you want to update.
5. Launch the `"docker-compose up -d"` command.



Note:

- Launching the `"docker-compose up -d"` command, the container is restarted and the database schema is automatically updated. If you are planning to update both the HCL Workload Automation server MDM and BKM, ensure that you run the command for one component at a time. To avoid database conflicts, start the second component only when the first component has completed successfully.
- In a Docker environment, if your server component uses a timezone different from the default timezone, then to avoid problems with the FINAL job stream, you must update `MAKEPLAN` within the `DOCOMMAND`, specifying the **timezone** parameter and value. For example, if you are using the America/Los Angeles timezone, then it must be specified as follows:

```
$JOBS

WA_WA-SERVER_XA#MAKEPLAN
DOCOMMAND "TODAY_DATE=`${UNISONHOME}/bin/datecalc today pic YYYYMMDD`; ${UNISONHOME}/MakePlan -to
`${UNISONHOME}/bin/datecalc ${TODAY_DATE}070
0 + 1 day + 2 hours pic MM/DD/YYYY^HHTT` timezone America/Los_Angeles"
STREAMLOGON wouser
DESCRIPTION "Added by composer."
TASKTYPE OTHER
SUCCOUTPUTCOND CONDSUCC "(RC=0) OR (RC=4)"
RECOVERY STOP
```

Only the following parameters can be modified with the update:

- DB_TYPE
- DB_HOSTNAME

- DB_PORT
- DB_NAME
- DB_TS_NAME
- DB_TS_PATH
- DB_LOG_TS_NAME
- DB_LOG_TS_PATH
- DB_PLAN_TS_NAME
- DB_PLAN_TS_PATH
- DB_TEMP_TS_NAME
- DB_SBSpace
- DB_USER
- DB_ADMIN_USER
- DB_SSL_CONNECTION
- WA_PASSWORD
- DB_ADMIN_PASSWORD
- DB_PASSWORD
- SSL_KEY_FOLDER
- SSL_PASSWORD

Updating containers when using default certificates

Updating the container configuration parameters when using default certificates.

Before you begin:

Modify the certificates as explained in the following procedure:

1. Access the server container.
2. Open the `localopts` file and check the certificates path in the following section:

```
SSL key      ="/home/wauser/wadata/FTAcert/TWSCClient.key"
SSL certificate ="/home/wauser/wadata/FTAcert/TWSCClient.cer"
SSL key pwd   ="/home/wauser/wadata/FTAcert/password.sth"
SSL CA certificate ="/home/wauser/wadata/FTAcert/TWSTrustCertificates.cer"
SSL random seed ="/home/wauser/wadata/FTAcert/TWS.rnd"
```

3. Exit the server container.
4. Copy all the certificates in a local directory by launching the following command: `docker cp`.
5. Rename the certificates as follows:

```
tls.key
tls.crt
tls.sth
ca.crt
tls.rnd
```

6. Ensure that in the `docker compose.yaml` file you have the following parameters for server, console, and agent components:

```
SSL_PASSWORD= default
SSL_KEY_FOLDER= <cert_directory>
```

where

<cert_directory> is the path of the directory where you saved the certificates.

7. Modify the volume `<path_on_host_containing_certs>:/opt/wautils/certs` with the path of the directory that contains your certificates at the place of `<path_on_host_containing_certs>`.

About this task:

To change the container configuration parameters or to obtain the latest version of a container, an update is required.

Complete the following procedure to update a Docker container:

1. Contact your HCL sales representative for the login details required to access the HCL Entitled Registry
2. Run the following command to log into the HCL Entitled Registry:

```
docker login -u <your_username> -p <your_entitled_key> hclcr.io
```

The console image is named `hclcr.io/wa/hcl-workload-automation-console:<release_name>`

3. Run the following command to log into the HCL Entitled Registry:

```
docker login -u <your_username> -p <your_entitled_key> hclcr.io
```

4. Manually update the compose file by modifying the *image* name if docker-compose does not reference the version to which you want to update.
5. Launch the `"docker-compose up -d"` command.



Note:

- Launching the `"docker-compose up -d"` command, the container is restarted and the database schema is automatically updated. If you are planning to update both the HCL Workload Automation server MDM and BKM, ensure that you run the command for one component at a time. To avoid database conflicts, start the second component only when the first component has completed successfully.
- In a Docker environment, if your server component uses a timezone different from the default timezone, then to avoid problems with the FINAL job stream, you must update `MAKEPLAN` within the `DOCOMMAND`, specifying the **timezone** parameter and value. For example, if you are using the America/Los Angeles timezone, then it must be specified as follows:

```
$JOBS

WA_WA-SERVER_XA#MAKEPLAN
DOCOMMAND "TODAY_DATE=`${UNISONHOME}/bin/datecalc today pic YYYYMMDD`; ${UNISONHOME}/MakePlan -to
`${UNISONHOME}/bin/datecalc ${TODAY_DATE}070
0 + 1 day + 2 hours pic MM/DD/YYYY^HHTT` timezone America/Los_Angeles"
STREAMLOGON wauser
DESCRIPTION "Added by composer."
TASKTYPE OTHER
```



```
SUCCOUTPUTCOND CONDSUCC "(RC=0) OR (RC=4)"  
RECOVERY STOP
```

Only the following parameters can be modified with the update:

- DB_TYPE
- DB_HOSTNAME
- DB_PORT
- DB_NAME
- DB_TS_NAME
- DB_TS_PATH
- DB_LOG_TS_NAME
- DB_LOG_TS_PATH
- DB_PLAN_TS_NAME
- DB_PLAN_TS_PATH
- DB_TEMP_TS_NAME
- DB_SBSPACE
- DB_USER
- DB_ADMIN_USER
- DB_SSL_CONNECTION
- WA_PASSWORD
- DB_ADMIN_PASSWORD
- DB_PASSWORD
- SSL_KEY_FOLDER
- SSL_PASSWORD

Chapter 21. FAQ - Upgrade procedures

A list of questions and answers related to upgrade procedures:

Q: How do I upgrade a component that was originally installed without SSL configuration?

A: To configure SSL attributes, perform the following steps:

1. Set the **security_level** parameter to **force_enabled** in the workstation definition and the **secureaddr** parameter to the secure port, as described in the section about configuring SSL attributes in *Administration Guide*.
2. Set the **nm SSL full port** parameter to the value of the secure port in the `localopts` file. For more information, see the topic about `localopts` details in *Administration Guide*

Q: How do I upgrade a component that was installed with default certificates?

A: Define the **JKS_SSL_PASSWORD** environment variable as described in the section about enhanced security with default certificates in *Overview*. For the full upgrade procedure, see [Upgrading on page 228](#). If you are using default certificates and want to install a new component to be connected to a back-level master, see [Upgrading in a mixed-version environment when using default certificates on page 371](#).

Q: What happens if I do not remember the password for the default certificates?

A: Before starting the upgrade, test the passwords for the certificates using the following keytool commands:

- ```
keytool -list -keystore TWSServerTrustFile.jks
-storepass my_password
```
- ```
keytool -list -keystore TWSServerKeyFile.jks  
-storepass my_password
```

Q: The upgrade failed because the password I provided for the certificates in the JKS_SSL_PASSWORD variable is incorrect. How can I recover from this error?

A. Before restarting the upgrade, perform the following steps:

1. Retrieve and test the password for the certificates, as described in [Q: What happens if I do not remember the password for the default certificates? on page 378](#)
2. Restore the previous version of the `ita.ini` file.
3. Restart the upgrade.

Part V. Enabling and disabling FIPS

FIPS is a U.S. government security standard that defines **security requirements for cryptographic modules** used to protect sensitive information. **FIPS 140-3** is the latest version of the U.S. and Canadian government security standard that defines security requirements for cryptographic modules in IT and telecommunications products. Its purpose is to ensure that products handling sensitive data via cryptography are secure and reliable. Federal agencies must use FIPS 140-3 validated modules. It is also widely adopted by defence contractors and financial institutions.

To ensure FIPS compliance, all HCL Workload Automation components must be at version 10.2.5 or later, certificates must employ at least a robust 2K RSA key and use encryption algorithms different from `MD5-RSA` and `SHA1-RSA`.

Read the following topics to find out how to enable FIPS in your environment:

- [Enabling FIPS at installation time on page 380](#)
- [Enabling or disabling FIPS at upgrade time on page 382](#)
- [Upgrading from a FIPS-enabled environment on page 385](#)
- [Enabling or disabling FIPS after installing or upgrading on page 387](#)



Note: FIPS 140-3 compliance: In agreement with the specifications provided in point 3 in [FIPS certified cryptography in IBM Semeru Runtimes](#), HCL Workload Automation operates under an exception regarding read access to PKCS#12 keystores by using a specific provider.

Chapter 22. Enabling FIPS at installation time

Quick and easy steps to enable FIPS when installing HCL Workload Automation for the first time.

About this task

If you are performing a fresh installation and want to enable FIPS while installing, perform the steps listed below on each component in the HCL Workload Automation environment.

To ensure FIPS compliance, all HCL Workload Automation components must be at version 10.2.5 or later, certificates must employ at least a robust 2K RSA key and use encryption algorithms different from `MD5-RSA` and `SHA1-RSA`.

When installing, you can encounter one of the following situations:

If certificates do not meet FIPS standards

An error message is displayed stating that the current security configuration does not support FIPS mode and the upgrade stops. To enable FIPS in full mode, proceed to step 1 on page 380.

If certificates meet FIPS standards

You can install and enable FIPS. Proceed to step 2 on page 380 onward.

1. Obtain secure certificates. You can also generate them using the `certman generate` command. For more information, see the topic about configuring secure communication in *Administration Guide*.
2. Start the installation on the master domain manager, as described in [Installing from the command-line interface on page 45](#), setting the **enablefips** parameter to `true`.
3. The installation completes, setting FIPS in **full** mode.
4. Check the version of the OpenSSL libraries present in your environment:
 - If the system provides **OpenSSL version 3.0 or higher**, those libraries are automatically used by the product.
 - If the system libraries do **not** meet the version requirement, the product defaults to using the **OpenSSL libraries included with HCL Workload Automation**.

If you are using the OpenSSL libraries provided with the operating system, set the machine in FIPS mode. Note that the specific command to enable this mode may differ depending on your operating system.

5. On the master domain manager, run the following commands to set the environment variables and check the security status:

```
./twc_env.sh  
secure -securitystatus
```

A message similar to the following is displayed:

```
FIPS is enabled on the master domain manager
```

Results

FIPS is now correctly enabled in **full** mode on the master domain manager.

Installing the Dynamic Workload Console in FIPS mode

About this task

To install the Dynamic Workload Console in FIPS mode, perform the following steps:

1. Install the Dynamic Workload Console, setting the **enablefips** parameter to `true`.
2. On the Dynamic Workload Console, run the following commands to set the environment variables and check FIPS status:

```
./dwc_env.sh
```

```
secure -securitystatus
```

A message similar to the following is displayed:

```
FIPS is enabled on the Dynamic Workload Console
```

Results

FIPS is now correctly enabled in **full** mode on the Dynamic Workload Console.

Installing agents in FIPS mode

About this task

To install the agents in FIPS mode, perform the following steps:

1. Install the agents, setting the **enablefips** parameter to `true`.
2. On each agent, run the following commands to set the environment variables and check FIPS status:

```
./twc_env.sh
```

```
secure -securitystatus
```

A message similar to the following is displayed:

```
FIPS is enabled on the agent
```

Results

FIPS is now correctly enabled in **full** mode on the agents.

Chapter 23. Enabling or disabling FIPS at upgrade time

Quick and easy steps to enable FIPS when upgrading from an environment where FIPS was not enabled.

About this task

If you are upgrading from an environment where FIPS was not enabled, and want to enable it while upgrading, perform the steps listed below on each component in the HCL Workload Automation environment.

To ensure FIPS compliance, all HCL Workload Automation components must be at version 10.2.5 or later, certificates must employ at least a robust 2K RSA key and use encryption algorithms different from `MD5-RSA` and `SHA1-RSA`.



Note: If your current environment is running any versions between 10.2.2 and 10.2.4, FIPS is disabled by default.

When upgrading, you can encounter one of the following situations:

If certificates do not meet FIPS standards

An error message is displayed stating that the current security configuration does not support FIPS mode and the upgrade stops. To enable FIPS, proceed to [step 1 on page 382](#).

If certificates meet FIPS standards

You can upgrade and enable FIPS. Proceed to [step 2 on page 382](#)

1. If your current certificates do not meet FIPS standards, replace them with CA-signed certificates, as explained in [Replacing Default SSL Certificates with CA signed Customer Certificates](#).
2. On the master domain manager, start the upgrade as described in [Upgrading from the CLI on page 233](#), setting the **enablefips** parameter to `true`.
3. The upgrade completes, enabling FIPS in **weak** mode. When in **weak** mode, the upgraded master domain manager can communicate with back-level components, ensuring business continuity.
4. Check the version of the OpenSSL libraries present in your environment:
 - If the system provides **OpenSSL version 3.0 or higher**, those libraries are automatically used by the product.
 - If the system libraries do **not** meet the version requirement, the product defaults to using the **OpenSSL libraries included with HCL Workload Automation**.

If you are using the OpenSSL libraries provided with the operating system, set the machine in FIPS mode. Note that the specific command to enable this mode may differ depending on your operating system.

5. On the master domain manager, run the following command to set the environment variables:

```
./twc_env.sh
```

6. On the master domain manager, run the following command to verify the security status:

```
secure -checksecurity
```

A message similar to the following is displayed:

```
FIPS configuration updated in weak mode. To enable full FIPS mode,
update the master domain manager and all backup master domain managers to the
current release. Then, run the secure -updatesecurity command on master domain manager.
```

As stated in the message, before you set up FIPS in full mode on the master domain manager, it is necessary to upgrade all components in your environment to version 10.2.5 or later.

7. Upgrade the remaining server components (backup master domain manager, dynamic domain manager, backup dynamic domain manager) if any, as described in [Upgrading from the CLI on page 233](#), setting the **enablefips** parameter to `true`.
8. Upgrade the Dynamic Workload Console, setting the **enablefips** parameter to `true`.
9. Upgrade the agents, setting the **enablefips** parameter to `true`.
10. On the master domain manager, run the following command to check the encryption level of user passwords in the database and change it from 3DES to AES, if necessary:

```
secure -updatesecurity
```

This command also sets the **useAESEncryptionAlgorithm** option to `yes`. For more information about global options, see the topics about global options in *Administration Guide*.

11. On the master domain manager and backup master domain manager, run the following command to set FIPS in **full** mode:

```
secure -fips on
```

The master domain manager switches to **full** FIPS mode **after** HCL Workload Automation processes are restarted. For more information about the `secure` command, see [Optional password encryption - secure script on page 427](#).

12. Restart the master domain manager and backup master domain manager to make the switch to **full** FIPS mode effective.
13. On the Dynamic Workload Console, run the following command to set the environment variables:

```
. ./dwc_env.sh
```

14. On the Dynamic Workload Console, run the following command to set FIPS in **full** mode:

```
secure -fips on
```

15. On each agent, run the following command to set the environment variables:

```
. ./twc_env.sh
```

16. On each agent, run the following command to set FIPS in **full** mode:

```
secure -fips on
```

17. Optionally run the following command to check FIPS status:

```
secure -securitystatus
```

A message similar to the following is displayed:

```
FIPS is enabled on the agent
```

18. **Restart all HCL Workload Automation processes on the Dynamic Workload Console and agents to make changes to FIPS configuration effective. To prevent communication problems after switching to full FIPS mode, ensure you perform a coordinated restart of the various components.**

Results

FIPS is now correctly enabled in **full** mode in your environment.

Disabling FIPS at upgrade time

About this task

FIPS mode is currently enabled in your source environment. You plan to upgrade to version 10.2.5 and disable FIPS during the upgrade process.

To ensure FIPS compliance, all HCL Workload Automation components must be at version 10.2.5 or later, certificates must employ at least a robust 2K RSA key and use encryption algorithms different from `MD5-RSA` and `SHA1-RSA`.

However, certificates in the source environment may not meet the security standards of FIPS 140-3, even if FIPS mode is currently enabled in the source environment.

If certificates are not secure by FIPS standard, the upgrade stops. To proceed with the upgrade, you can either obtain secure certificates, as described in [Upgrading from a FIPS-enabled environment on page 385](#), or, if FIPS compliance is not required, you can restart the upgrade setting the **enablefips** parameter to `false` when upgrading each component.

Results

After you have upgraded all components setting the **enablefips** parameter to `false`, FIPS is disabled in your upgraded environment.

Chapter 24. Upgrading from a FIPS-enabled environment

Upgrading from a FIPS-enabled environment

About this task

To ensure FIPS compliance, all HCL Workload Automation components must be at version 10.2.5 or later, certificates must employ at least a robust 2K RSA key and use encryption algorithms different from `MD5-RSA` and `SHA1-RSA`.

When upgrading from an environment where FIPS is enabled, you can encounter one of the following situations:

If certificates do not meet FIPS standards

An error message is displayed stating that the current security configuration does not support FIPS mode and the upgrade stops. To enable FIPS in full mode, proceed to [step 1 on page 385](#).

If certificates meet FIPS standards

You can upgrade and maintain FIPS enabled. Proceed to [step 2 on page 385](#).

To make your environment FIPS compliant, perform the procedure described below on all components in your environment.

1. If your current certificates do not meet FIPS standards, replace them with CA-signed certificates, as explained in [Replacing Default SSL Certificates with CA signed Customer Certificates](#).
2. On the master domain manager, start the upgrade procedure, as described in [Upgrading from the CLI on page 233](#). HCL Workload Automation discovers that FIPS is enabled in the source environment and proceeds with enabling it in the target environment.
3. The upgrade completes, enabling FIPS in **weak** mode. When in **weak** mode, the upgraded master domain manager can communicate with back-level components, ensuring business continuity.
4. On the master domain manager, run the following command to set the environment variables:

```
. ./twc_env.sh
```

5. On the master domain manager, run the following command to verify the security status:

```
secure -checksecurity
```

A message similar to the following is displayed:

```
FIPS configuration updated in weak mode. To enable full FIPS mode,
update the master domain manager and all backup master domain managers to the
current release. Then, run the secure -updatesecurity command on master domain manager.
```

As stated in the message, before you set up FIPS in full mode on the master domain manager, it is necessary to upgrade all components in your environment to version 10.2.5 or later.

6. Upgrade the remaining server components (backup master domain manager, dynamic domain manager, backup dynamic domain manager) if any, as described in [Upgrading from the CLI on page 233](#).
7. Upgrade the Dynamic Workload Console.
8. Upgrade the agents.
9. On the master domain manager, run the following command to check the encryption level of user passwords in the database and change it from 3DES to AES, if necessary:

```
secure -updatesecurity
```

This command also sets the **useAESEncryptionAlgorithm** option to `yes`. For more information about global options, see the topics about global options in *Administration Guide*.

10. On the master domain manager and backup master domain manager, run the following command to set FIPS in **full** mode:

```
secure -fips on
```

The master domain manager switches to **full** FIPS mode **after** HCL Workload Automation processes are restarted. For more information about the `secure` command, see [Optional password encryption - secure script on page 427](#).

11. Restart the master domain manager and backup master domain manager to make the switch to **full** FIPS mode effective.
12. On the Dynamic Workload Console, run the following command to set the environment variables:

```
./dwc_env.sh
```

13. On the Dynamic Workload Console, run the following command to set FIPS in **full** mode:

```
secure -fips on
```

14. On each agent, run the following command to set the environment variables:

```
./twc_env.sh
```

15. On each agent, run the following command to set FIPS in **full** mode:

```
secure -fips on
```

16. **Restart all HCL Workload Automation processes on the Dynamic Workload Console and agents to make changes to FIPS configuration effective. To prevent communication problems after switching to full FIPS mode, ensure you perform a coordinated restart of the various components.**

Results

FIPS is now correctly enabled in **full** mode in your environment.

Chapter 25. Enabling or disabling FIPS after installing or upgrading

You can easily enable FIPS after completing the installation or upgrade.

About this task

To ensure FIPS compliance, all HCL Workload Automation components must be at version 10.2.5 or later, certificates must employ at least a robust 2K RSA key and use encryption algorithms different from `MD5-RSA` and `SHA1-RSA`.

When installing, you can encounter one of the following situations:

If certificates do not meet FIPS standards

An error message is displayed stating that the current security configuration does not support FIPS mode and the upgrade stops. To enable FIPS in full mode, proceed to [step 1 on page 387](#).

If certificates meet FIPS standards

You can install and enable FIPS. Proceed to [step 2 on page 387](#).

To enable FIPS after completing the installation or upgrade, perform the following steps:

1. If your current certificates do not meet FIPS standards, replace them with CA-signed certificates, as explained in [Replacing Default SSL Certificates with CA signed Customer Certificates](#).
2. Check the version of the OpenSSL libraries present in your environment:
 - If the system provides **OpenSSL version 3.0 or higher**, those libraries are automatically used by the product.
 - If the system libraries do **not** meet the version requirement, the product defaults to using the **OpenSSL libraries included with HCL Workload Automation**.

If you are using the OpenSSL libraries provided with the operating system, set the machine in FIPS mode. Note that the specific command to enable this mode may differ depending on your operating system.

3. On the master domain manager, run the following command to set the environment variables:

```
./twc_env.sh
```

4. On the master domain manager, run the following command to check the encryption level of user passwords in the database and change it from 3DES to AES, if necessary:

```
secure -updatesecurity
```

This command also sets the `useAESEncryptionAlgorithm` option to `yes`. For more information about global options, see the topics about global options in *Administration Guide*.

5. On the master domain manager and backup master domain manager, run the following command to set FIPS in **full** mode:

```
secure -fips on
```

The master domain manager switches to **full** FIPS mode **after** HCL Workload Automation processes are restarted.

For more information about the `secure` command, see [Optional password encryption - secure script on page 427](#).

6. On the Dynamic Workload Console, run the following command to set the environment variables:

```
./dwc_env.sh
```

7. On the Dynamic Workload Console, run the following command to set FIPS in **full** mode:

```
secure -fips on
```

8. On each agent, run the following command to set the environment variables:

```
. ./twc_env.sh
```

9. On each agent, run the following command to set FIPS in **full** mode:

```
secure -fips on
```

10. **Restart all HCL Workload Automation processes on the Dynamic Workload Console and agents to make changes to FIPS configuration effective. To prevent communication problems after switching to full FIPS mode, ensure you perform a coordinated restart of the various components.**

Results

You have now enabled FIPS in **full** mode in your environment.

Disabling FIPS after installing or upgrading

About this task

You can easily disable FIPS after completing the installation or upgrade.

To disable FIPS after completing the installation or upgrade, perform the following steps:

1. Set the environment variables on all components.
2. Run the following command on all components in your environment to disable FIPS:

```
secure -fips off
```

FIPS is disabled **after** HCL Workload Automation processes are restarted.

3. **Restart all HCL Workload Automation processes. To prevent communication problems after disabling FIPS, ensure you perform a coordinated restart of the various components.**

Part VI. Moving your workload from an on-premises to a cloud environment

A quick procedure to move your workload from an on-premises to a cloud environment

About this task

Moving your workload from an on-premises to a cloud environment is a quick procedure which involves configuring SSL communication between your existing on-premises master domain manager and a new backup master domain manager on the cloud. You then switch permanently domain management capabilities from the on-premises master domain manager to the backup master domain manager on the cloud to shift your whole workload to the cloud. This procedure requires the on-premises master domain manager to be at Version 9.5 Fix Pack 3 or later.

At the end of the procedure, you will have switched your master domain manager to the cloud and set up your dynamic agents to work in SSL mode with the on-cloud master domain manager

This procedure applies to the following clusters:

Amazon Elastic Kubernetes Service (EKS)

For this cluster, you can use an ingress-type network or a load-balancer network. To specify which network type you want to use, set the relevant parameters in the `values.yaml` file. For detailed information, see the **Network enablement** section in [HCL Workload Automation](#).

OpenShift

For this cluster, you can only use routes as network service. An OpenShift Container Platform route allows you to associate a service with an externally-reachable host name. This edge host name is then used to route traffic to the service. For more information, see the readmes available in the section about Deploying product components on Red Hat OpenShift in *Planning and Installation Guide*.



Note: On-premises fault-tolerant agents cannot connect to an on-cloud master domain manager.

On-premises side operations

Before you begin

Ensure the following conditions are met for your on-premises master domain manager:

- Version 9.5, Fix Pack 3 or later is installed.
- The port number used by the `netman` process to listen for communication from the dynamic domain manager (**brnetmanport**) is set to the default **41114** value.
- Ensure the `SECURITYLEVEL` attribute is set to `force`, or `force_enabled`. For more information about workstation definition parameters, see the section about workstation definition in *User's Guide and Reference*.

About this task

Perform the following operations on the on-premises side:

1. Set the HCL Workload Automation environment variables:

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. Configure your master domain manager for SSL communication using the modify command:

```
composer modify ws your_master_domain_manager
```

- a. In the **secureaddr** argument, define the port used to listen for incoming SSL connections, for example 31113 or another available port.
- b. In the **securitylevel** argument, specify `enabled` to set the master domain manager to uses SSL authentication only if its domain manager workstation or another fault-tolerant agent below it in the domain hierarchy requires it.

See the following example:

```
CPUNAME your_mdm_name
DESCRIPTION "MANAGER CPU"
OS UNIX
NODE your_IP_address TCPADDR 31111
SECUREADDR 31113
DOMAIN MASTERDM
FOR MAESTRO
  TYPE MANAGER
  AUTOLINK ON
  BEHINDFIREWALL OFF
SECURITYLEVEL ENABLED
  FULLSTATUS ON
END
```

For more information about the modify command and the workstation definition, see the related sections in *User's Guide and Reference*.

3. Modify the `localopts` file to enable SSL communication, as follows:

- a. Browse to the `TWA_DATA_DIR` folder.
- b. Edit the following properties in the `localopts` file. See the following example:

```
nm SSL full port      =0
nm SSL port          =31113
SSL key               ="/install_dir/ssl/OpenSSL/TWSCClient.key"
SSL certificate       ="/install_dir/ssl/OpenSSL/TWSCClient.cer"
SSL key pwd           ="/install_dir/ssl/OpenSSL/password.sth"
SSL CA certificate    ="/install_dir/ssl/OpenSSL/TWSTrustCertificates.cer"
SSL random seed       ="/install_dir/ssl/OpenSSL/TWS.rnd"
```

where:

nm SSL port

Is the port used to listen for incoming SSL connections, when full SSL is not configured, for example 31113.

For more information about the `localopts` file, see the section about setting local options in *Administration Guide*.

4. If you have a dynamic domain manager in your environment, repeat steps 2 on page 390 and 3 on page 390 on the dynamic domain manager to have the dynamic domain manager function correctly with the on-cloud master domain manager. The dynamic domain manager stays in the on-premises environment.
5. If you want to use custom SSL certificates, edit the paths in the `localopts` file specifying the paths to the custom certificates and using the same names as the default certificates. For more information about secure connections, see the section about configuring secure communications in *Administration Guide*.
6. Stop HCL Workload Automation Batchman process by running this command:

```
conman stop
```

7. Stop HCL Workload Automation Netman process by running this command:

```
conman shut
```

8. Restart HCL Workload Automation processes by running these commands:

```
StartUp
```

```
conman start
```

9. You can optionally configure your on-premises fault-tolerant agents for communicating with the on-cloud master domain manager, by performing this procedure on each fault-tolerant agent.

Cloud-side operations

Before you begin

If you are using OpenShift, the connection between the on-premises master domain manager and the on-cloud backup master domain manager takes place through routes; therefore, it is recommended to use short names for namespaces, especially if the cluster name is long. This is because workstation host names cannot exceed 51 characters, therefore, the route must comply with this maximum character length.

About this task

Perform the following operations on the cloud side:

1. Download the latest product version. See
 - If you are using Amazon EKS, see [HCL Workload Automation](#) for information about downloading images, installing, and configuring the product.
 - If you are using OpenShift, see [Deploying HCL Workload Automation components on Red Hat OpenShift using helm charts on page 176](#).
2. Open the `values.yaml` file to configure a new server instance.

If you want to deploy only a new server without the Agent and Console applications, set the **enableAgent** and **enableConsole** parameters to `false`.

3. Set the following database parameters to have the new server instance point the database of the on-premises master domain manager. These values must match the values defined for the on-premises master domain manager.

```
db:
  adminUser: <admin_dbuser>
  hostname: <db_host>
  name: <db_name>
  port: <db_port>
  sslConnection: false
  tsName: null
  tsPath: null
  tsTempName: null
  tssbspace: null
  type: <db_type>
  usepartitioning: true
  user: <db_user>
```

This automatically configures the on-cloud server as a backup master domain manager for the on-premises master domain manager.

4. Set the **server.enableSingleInstanceNetwork** parameter to `true` to create an additional load balancer for each server pod. This is used to connect the backup master domain manager inside the cluster with master domain manager outside the cluster. For more information about parameters, see the **Configuration Parameters** section in [HCL Workload Automation](#).
5. To deploy the new server instance in a cloud environment, `type`:

```
helm install -f values.yaml workload_automation_release_name workload/hcl-workload-automation-prod
-n workload_automation_namespace
```

where:

workload_automation_release_name

is the name of the release, for example `hwa`.

Result

When you deploy the backup master domain manager on the cloud, it is automatically configured as follows, in full SSL mode with the on-premises master domain manager:

```
CPUNAME HWA-SERVER-0
DESCRIPTION "FTA CPU"
OS UNIX
NODE hwa-waserver-0.hwa-test TCPADDR 31111
SECUREADDR 443
DOMAIN MASTERDM
FOR MAESTRO
TYPE FTA
AUTOLINK ON
BEHINDFIREWALL OFF
SECURITYLEVEL FORCE_ENABLED
FULLSTATUS ON
END
```

where

hwa-waserver-0.hwa-test

Is the name of the ingress-type network being configured, if you are using an ingress-type network for EKS.

If you are using a load-balancer network, the `NODE` parameter is automatically set to the IP address of the load balancer. For more information, see the **Network enablement** section in [HCL Workload Automation](#).

If you are deploying on OpenShift, this parameter is automatically set to the OpenShift network route. For more information, see the readmes available in [Deploying product components on Red Hat OpenShift, V4.x](#).

SECURITYLEVEL

Specifies the type of SSL authentication for the workstation. This parameter is automatically set to `force_enabled`, which means that the workstation uses SSL authentication for all of its connections to all target workstations which are set to this value. The workstation tries to establish a connection in FULLSSL mode and, if the attempt fails, it tries to establish an unsecure connection. For more information about workstation definition parameters, see the section about workstation definition in *User's Guide and Reference*.

In the same way, the `localopts` file of the backup master domain manager on the cloud is also automatically configured for SSL communication. See the following example:

```
nm SSL full port      =31113
#
nm SSL port          =0
#
SSL key   ="/home/wauser/wadata/FTAcert/TWSCClient.key"
SSL certificate ="/home/wauser/wadata/FTAcert/TWSCClient.cer"
SSL key pwd  ="/home/wauser/wadata/FTAcert/password.sth"
SSL CA certificate ="/home/wauser/wadata/FTAcert/TWSTrustCertificates.cer"
SSL random seed ="/home/wauser/wadata/FTAcert/TWS.rnd"
```

6. To assign full control for all objects to the **wauser**, type the following command:

```
composer mod acl @
```

The following example shows the modified access control list:

```
ACCESSCONTROLLIST FOR ALLOBJECTS
  root FULLCONTROL
  twsuser FULLCONTROL
  wauser FULLCONTROL
END
```

```
ACCESSCONTROLLIST FOLDER /
  root FULLCONTROL
  twsuser FULLCONTROL
  wauser FULLCONTROL
END
```

Switching domain manager capabilities

About this task

Final steps to switch domain manager capabilities permanently

1. To switch the event processor, run the following command either on the master domain manager or backup master domain manager:

```
switcheventprocessor [folder/]workstation
```

For more information about the command, see the section about the switcheventprocessor command in *User's Guide and Reference*.

2. To switch domain management capabilities, run the following command either on the master domain manager or backup master domain manager:

```
switchmgr domain;newmgr
```

For more information about the command, see the section about the switchmgr command in *User's Guide and Reference*.

3. To make the switch permanent, edit from composer the definition of the previous master domain manager. See the following example and notice how the **TYPE** attribute changes from `MANAGER` to `FTA`.

PREVIOUS DEFINITION

```
CPUNAME your_mdm_name
DESCRIPTION "MANAGER CPU"
OS UNIX
NODE your_IP_address TCPADDR 31111
SECUREADDR 31113
DOMAIN MASTERDM
FOR MAESTRO
TYPE MANAGER
AUTOLINK ON
BEHINDFIREWALL OFF
SECURITYLEVEL ENABLED
FULLSTATUS ON
END
```

NEW DEFINITION

```
CPUNAME your_mdm_name
DESCRIPTION "MANAGER CPU"
OS UNIX
NODE your_IP_address TCPADDR 31111
SECUREADDR 31113
DOMAIN MASTERDM
FOR MAESTRO
TYPE FTA
AUTOLINK ON
BEHINDFIREWALL OFF
SECURITYLEVEL ENABLED
FULLSTATUS ON
END
```

4. To make the switch permanent, edit from composer the definition of the previous backup master domain manager. See the following example and notice how the **TYPE** attribute changes from **FTA** to **MANAGER**.

PREVIOUS DEFINITION

```
CPUNAME HWA-SERVER-0
DESCRIPTION "FTA CPU"
OS UNIX
NODE hwa-waserver-0.hwa-test TCPADDR 31111
SECUREADDR 443
DOMAIN MASTERDM
FOR MAESTRO
  TYPE FTA
  AUTOLINK ON
  BEHINDFIREWALL OFF
  SECURITYLEVEL FORCE_ENABLED
  FULLSTATUS ON
END
```

NEW DEFINITION

```
CPUNAME HWA-SERVER-0
DESCRIPTION "FTA CPU"
OS UNIX
NODE hwa-waserver-0.hwa-test TCPADDR 31111
SECUREADDR 443
DOMAIN MASTERDM
FOR MAESTRO
  TYPE MANAGER
  AUTOLINK ON
  BEHINDFIREWALL OFF
  SECURITYLEVEL FORCE_ENABLED
  FULLSTATUS ON
END
```

5. To make the changes effective, run the following command:

```
JnextPlan -for 0000
```

6. Optionally, you can deploy a new backup master domain manager on the cloud by performing a scale-up of the components listed in the `values.yaml` file. To perform this operation, set the **waserver.replicaCount** parameter to a value higher than 1. You can now optionally uninstall your on-premises backup master domain manager.
7. To edit the FINAL and FINALPOSTREPORT job streams, type the following command:

```
composer mod js your_xa#final@ full
```

where:

your_xa

is the name of the extended agent workstation installed with the master domain manager.

Edit the following section:

```
STREAMLOGON old_tws_user
```

as follows:

```
STREAMLOGON wauser
```

8. Delete the FINAL and FINALPOSTREPORTS job streams from the plan, as follows:

```
conman "canc your_xa#FINALPOSTREPORTS"
```

```
conman "canc your_xa#FINAL"
```

9. Submit first the FINAL, and then the FINALPOSTREPORTS job streams into the current plan, as follows:

```
conman sbs your_xa#FINAL
```

```
conman sbs your_xa#FINALPOSTREPORTS
```

10. Reset the value of the **limit** job stream keyword for the FINAL and FINALPOSTREPORTS job streams, both in the database and in the plan, as follows:

```
conman "limit your_xa#FINAL ;10"
```

```
conman "limit your_xa#FINALPOSTREPORTS ;10"
```

11. To have your dynamic agents connect to the on-cloud master domain manager, copy the certificates located in /home/wauser/wadata/ITA/cpa/ita/cert/ and duplicate them to /datadir/ITA/cpa/ita/cert. Perform this operation for each on-premises dynamic agent in your environment.

Result

You have now successfully switched your master domain manager to the cloud and set up your dynamic agents to work in SSL mode with the on-cloud master domain manager.

Part VII. Troubleshooting installation, migration, and uninstallation

An overview on troubleshooting installation, migration, and uninstallation of the HCL Workload Automation.

Issues dealing with the installation, removal, and configuration of HCL Workload Automation and its prerequisites.

For information about issues on the DB2® installation, see the DB2® product documentation.

Installation log files

The type of log files you find on your system depends on the type of installation you performed.

On UNIX operating systems, the storage of data generated by HCL Workload Automation, such as logs and configuration files, are stored by default in the `DATA_DIR` directory, which you can optionally customize at installation time. By default, this directory is `<TWA_home>/TWSDATA` for the server and agent components, and `<DWC_home>/DWC_DATA` for the Dynamic Workload Console. The product binaries are stored instead, in the installation directory. For more information, see [Server components installation - serverinst script on page 442](#), [Dynamic Workload Console installation - dwcinst script on page 456](#), and [Agent installation parameters - twsinst script on page 119](#).



Note: If you deployed the product components using Docker containers, this is the default behavior and it cannot be modified. However, if you installed the product components using the command-line installation, the `--data_dir` parameter can be used to change the path.

master domain manager or dynamic domain manager and its backup

```
<TWA_home>/TWSDATA/installation/logs
```

Dynamic Workload Console

```
<DWC_home>/DWC_DATA/installation/logs
```

Dynamic agents and fault-tolerant agents

```
<INST_DIR>/TWSDATA/installation/logs/  
twsinst_<operating_system>_<TWS_user>^<product_version_number>.log. For more  
information, see The twsinst log files on page 399.
```

On Windows operating systems, installation log files are stored in the following paths:

master domain manager or dynamic domain manager and its backup

```
<INSTALL_DIR>\logs
```

Dynamic Workload Console

```
<INSTALL_DIR>\logs
```

When you install a fix pack, the suffix at the end of the file name lists the fix pack number in addition to the General Availability version number, for example:

```
serverinst_<version_number>.0.0<fix_pack_number>.log
```

Chapter 26. The twsinst log files

About this task

The twsinst log file name is:

On Windows operating systems:

```
<TWS_INST_DIR>\logs\twsinst_operating_system_TWS_user^version_number.log
```

Where:

TWS_INST_DIR

The HCL Workload Automation installation directory. The default installation directory is C :

\Program Files\HCL\TWA_TWS_user.

operating_system

The operating system.

TWS_user

The name of the user for which HCL Workload Automation was installed, that you supplied during the installation process.

On UNIX operating systems:

```
<TWS_INST_DIR>/TWSDATA/installation/logs/  
twsinst_operating_system_TWS_user^product_version_number.log
```

Where:

TWS_INST_DIR

The HCL Workload Automation installation directory. The default installation directory is /opt /

HCL/TWA_TWS_user.

operating_system

The operating system.

TWS_user

The name of the user for which HCL Workload Automation was installed, that you supplied during the installation process.

Chapter 27. Analyzing return codes for agent installation, upgrade, restore, and uninstallation

Check how your operation completed by analyzing the return codes that are issued by twsinst.

Return codes that you can receive when you are installing, upgrading, restoring, or uninstalling agents. To analyze them and take corrective actions, run the following steps:

On Windows operating systems

1. Display the operation completion return code, by using the following command:

```
echo %ERRORLEVEL%
```

2. Analyze the following table to verify how the operation completed:

Table 23. Windows operating system agent return codes

Error Code	Description	User action
0	Success: The operation completed successfully without any warnings or errors.	None.
1	Generic failure	Check the messages that are displayed on the screen by the script. Correct the error and rerun the operation. If the error persists, contact Support .
2	The installation cannot create the HCL Workload Automation user or assign the correct permission to it.	Verify the operating system policies and configuration. Verify the input values. If necessary, create the user manually before you run the installation.
3	The password is not correct or the installation cannot verify it.	Verify the operating system policies and configuration. Verify the input values.
4	The HCL Workload Automation installation directory is not empty. You specified as installation folder a directory that exists.	Empty it or specify a different directory.
5	An error occurred checking the HCL Workload Automation prerequisites on the workstation.	See the System Requirements Document at HCL Workload Automation Detailed System Requirements .
6	The HCL Workload Automation registry is corrupted.	Use the recovInstReg option to recover the registry. Then, rerun the operation.
7	The upgrade or restore operation cannot retrieve the information from the configuration files.	Check that the previous installation and the <code>localopts</code> , the <code>globalopts</code> , the

Error Code	Description	User action
		ita.ini, and the JobManager.ini files are not corrupted. Correct the errors and try again the operation.
8	The upgrade, restore, or uninstallation cannot proceed because there are jobs that are running.	Stop the jobs that are running or wait for these jobs to complete. Restart the operation.
9	The upgrade, restore, or uninstallation cannot proceed because there are files that are locked.	Stop all the processes that are running and close all the activities that can block the installation path. Restart the operation.
10	The upgrade, restore, or uninstallation cannot proceed because there are command lines opened.	Close the command lines. Restart the operation.

On UNIX and Linux operating systems:

1. Display the installation completion return code, by using the following command:

```
echo $?
```

2. Analyze the following table to verify how the installation completed:

Table 24. UNIX or Linux operating system agent return codes

Error Code	Description	User action
0	Success: The installation completed successfully without any warnings or errors.	None.
1	Generic failure.	Check the messages that are displayed on the video by the script. Correct the error and rerun the operation. If the error persists, contact Support .
2	The installation did not find the HCL Workload Automation user or its home directory. The HCL Workload Automation user that you specified either does not exist or does not have an associated home directory.	Verify the operating system definition of the HCL Workload Automation user.
3	Not applicable	

Error Code	Description	User action
4	The HCL Workload Automation installation directory is not empty. You specified as installation folder a directory that exists.	Empty it or specify a different directory.
5	An error occurred checking the HCL Workload Automation prerequisites on the workstation.	See the System Requirements Document at HCL Workload Automation Detailed System Requirements .
6	The HCL Workload Automation registry is corrupted.	Use the <code>recovInstReg</code> option to recover the registry. Then, rerun the operation.
7	The upgrade or restore operation cannot retrieve the information from the configuration files.	Check that the previous installation and the <code>localopts</code> , the <code>globalopts</code> , the <code>ita.ini</code> , and the <code>JobManager.ini</code> files are not corrupted. Correct the errors and try again the operation.
8	The upgrade, restore, or uninstallation cannot proceed because there are jobs that are running.	Stop the jobs that are running or wait for these jobs to complete. Restart the operation.
9	The upgrade, restore, or uninstallation cannot proceed because there are files that are locked.	Stop all the processes that are running and close all the activities that can block the installation path. Restart the operation.
10	The upgrade, restore, or uninstallation cannot proceed because there are command lines opened.	Close the command lines. Restart the operation.

Chapter 28. Problem scenarios: install, reinstall, upgrade, migrate, and uninstall

Known problems and troubleshooting

This section describes known problem scenarios that could occur with the installation, re-installation, upgrade, migration, and uninstallation of HCL Workload Automation components.

Installation or upgrade fails on RHEL version 9 and later

Installing or upgrading on RHEL version 9 and later fails if you were using default certificates.

About this task

Problem scenario

You are using a product version earlier than 10.2.1 with default certificates and you plan to upgrade to version 10.2.5, or you have upgraded to version 10.2.1 with default certificates and now plan to upgrade to 10.2.5. This problem can also occur if you perform a parallel upgrade from versions 9.4 or 9.5, which require a fresh installation of HCL Workload Automation components. If one or more HCL Workload Automation components are installed on RHEL version 9 or later, the upgrade or fresh installation fails.

You might encounter an error message similar to the following:

```
AWSRES003E The REST service cannot be contacted. Check if the service is
running or the existence of firewall rules or some issues on the dns side resolving
the server hostname that could prevent contacting the service.
```

Cause and solution

The SHA-1 signatures contained in the HCL Workload Automation default certificates are not supported by the OpenSSL libraries embedded in RHEL version 9 or later. This is a known problem with RHEL version 9 and later. For more information, see [Bug 2055796 - Enable SHA-1 signatures through LEGACY policy configuration](#).

To work around this problem, perform the following steps:

1. Stop all HCL Workload Automation services and WebSphere Application Server Liberty, by running the following commands:

```
conman stop; wait
conman shut; wait
conman ShutDownLwa
stopappserver
```

2. Browse to the following paths:

On UNIX™ operating systems

`TWA_DATA_DIR\ssl`

On Windows™ operating systems

`installation_dir\TWS\ssl`

3. Edit the `openssl.cnf` file as follows:

- add the **`alg_section = evp_properties`** property in section **`[openssl_init]`**.
- create a new section named **`[evp_properties]`** with this content:

```
#to enable in RHEL-9 using the embedded OpenSSL 3.0.x the support of SHA-1
#for signature creation and verification
rh-allow-sha1-signatures = yes
```

4. Restart all HCL Workload Automation and WebSphere Application Server Liberty services by running the following commands:

```
conman start
conman startappserver
StartUpLwa
```

Installing or linking a fault-tolerant agent earlier than 10.2 in an environment configured with new default or new custom certificates

About this task

If you install or link a fault-tolerant agent earlier than V10.2 to an environment with a master domain manager at version 10.2.1 or later using new custom or new default certificates, you need to manually install the certificates on the fault-tolerant agent. To perform the manual installation, you can follow the steps below:

1. Shut down the workstation of the fault-tolerant agent
2. Create a backup of the `localopts` file
3. Create a new folder under the target fault-tolerant agent as in the following example: `<TWS_DATA_DIR>/<new_cert_folder>`
4. From the master domain manager, copy the files under `TWA_DATA_DIR/ssl/depot`
5. Paste the `TWA_DATA_DIR/ssl/depot` files from the master domain manager into the `<TWS_DATA_DIR>/<new_cert_folder>` new directory
6. Update the `localopts` file of the fault-tolerant agent by changing the values of the following properties:
 - **SSL Key:** `<TWS_DATA_DIR>/<new_cert_folder>/tls.key`
 - **SSL Certificate:** `<TWS_DATA_DIR>/<new_cert_folder>/tls.crt`
 - **SSL key pwd:** `<TWS_DATA_DIR>/<new_cert_folder>/tls.sth`
 - **SSL CA certificate:** `<TWS_DATA_DIR>/<new_cert_folder>/ca.crt`
7. Turn on the workstation of the fault-tolerant agent.

Dynamic agents not connecting after certificate rotation when using JWT

About this task

If you are using JWT and modify the certificates on the master domain manager, communication with dynamic agents might be interrupted.

Workaround

After modifying the certificates on the master domain manager, if the communication has been lost, you can recover it by running the following command on each dynamic agent:

```
./AgentCertificateDownloader.sh --jwt false --work_dir <work_dir> --tdwbhostname <tdwbhostname>
--tdwbport <tdwbport> --gwid <gateway_id> --gateway <local|remote|none>
--apikey <API key for authentication with the master domain manager>
```

After running the command, restart the agent.

Problem prevention

If you are using JWT, you can disable this setting before modifying the certificates to prevent communication problems. To disable JWT and switch to using certificates, perform the following steps:

1. Browse to the `ita.ini` file on the dynamic agent. The file is located in `TWA_DATA_DIR/ITA/cpa/ita`.
2. Comment out the `jwt_file =` line, if existing. Consider the following example:

```
#jwt_file =
```

3. Restart the agent.

Uninstallation of Dynamic Workload Console fails

You are trying to uninstall an instance of the Dynamic Workload Console which is installed in a subfolder of the master domain manager installation directory.

About this task

As stated in [Installing the Dynamic Workload Console servers on page 110](#), the Dynamic Workload Console cannot be installed in a folder or subfolder containing another instance of an HCL Workload Automation component.

If you are trying to uninstall a Dynamic Workload Console installed in a nested path, you might encounter a generic error message.

To work around this problem, perform the following steps:

1. Stop WebSphere Application Server Liberty Base as described in the topic about starting and stopping the application server in *Administration Guide*.
2. Delete the Dynamic Workload Console directory.
3. Delete the registry files located in one of the following paths, depending on your operating system:

On Windows operating systems

```
/Windows/TWA
```

On UNIX operating systems

```
/etc/TWA
```

4. Restart WebSphere Application Server Liberty Base as described in the topic about starting and stopping the application server in *Administration Guide*.

What to do next

You can now proceed to install the Dynamic Workload Console in a different path.

Error in testing a connection or running reports on an engine returned from Fix Pack 1 to GA level when using an MSSQL database

If you install General Availability (GA) version 9.5 and a fix pack on a master domain manager using an MSSQL database and then return the workstation to GA version 9.5, you might experience problems when testing the engine connection and running reports.

When you try to test the engine connection or run a report, the operation fails and the following messages are displayed in the Dynamic Workload Console:

- AWSUI0803W Test connection to *engine_name*: engine successful, database failed.
- AWSUI0360E The JDBC URL is not configured on the selected engine, so the reporting capabilities cannot be used. Contact the IBM Workload Scheduler administrator.

Cause and solution:

The reporting feature for the MSSQL databases is released with version 9.5, Fix Pack 1. If you return the master domain manager to the GA version, you can no longer use the reporting feature for the MSSQL databases. To continue working with the Dynamic Workload Console, disable the database configuration for the reporting feature by performing the following steps:

1. Log in to the Dynamic Workload Console and select Administration > Manage Engines.
2. Click on the engine you returned to the GA version.
3. In the **Database Configuration for Reporting** section, disable the **Enable Reporting** check box.

Error in upgrading the HCL Workload Automation database when using a DB2 database

When you run the configureDB script to upgrade DB2 when upgrading to HCL Workload Automation 9.5 or later, the following error messages are returned:

- ALTER TABLE LOG.LLRC_LOG_RECORDS ADD COLUMN LLRC_DIFFERENCE VARCHAR (4095) DB21034E. The command was processed as an SQL statement because it was not a valid Command Line Processor command.
- QL0670N The statement failed because the row or column size of the resulting table would have exceeded the row or column size limit: "8101". Table space name: "LOG_DAT_8K". Resulting row or column size: "10000". SQLSTATE=54010

Cause and solution:

If you try to upgrade HCL Workload Automation to version 9.5 or later, and the HCL Workload Automation database was created with DB2, the DB2 option **EXTENDED_ROW_SZ** remains set to DISABLE during the upgrade process.

Starting from HCL Workload Automation version 9.5, the LOG. LLRC_LOG_RECORDS table exceeds the table space or buffer pool page size which was previously set to 8 kilobytes and this causes the upgrade process to fail.

You can solve the problem by either changing the EXTENDED_ROW_SZ DB2 configuration parameter or, if you do not want to change this parameter, migrate the tables to a new buffer pool and table space with a page size of 16 kilobytes:

Change the DB2 configuration parameter

Change the DB2 configuration parameter EXTENDED_ROW_SZ to ENABLE.

OR

Create a new buffer pool and table space and migrate the tables to the new table space

1. Create a new buffer pool and table space with a page size of 16 kilobytes instead of 8 kilobytes.
2. Migrate the involved tables, which are defined in the LOG schema, to the new table space.

Problems in encrypting the useropts file

About this task

You have upgraded from a version earlier than 10.2 with encryption automatically enabled, but the `useropts` file is not encrypted.

To solve this problem, launch the following command on the master domain manager:

```
UpdateUseropts -update twsuser twsuserpassword
```

where:

twsuser

is the name of the user whose password you want to encrypt.

twsuserpassword

is the password you want to encrypt.

The `useropts` file is encrypted immediately.

For more information about the `useropts` file, see the section about setting user options in *Administration Guide*.

For more information about enabling product encryption after upgrading, see the related section in *HCL Workload Automation: Planning and Installation*.

WebSphere Application Server Liberty server does not start when applying a fix pack to the backup master domain manager

A failure occurs when applying version 9.5, Fix Pack 4, or later, to a previous fix pack.

If the upgrade process fails starting the Liberty application server, with a message similar to the following:

```

WAINST200I Configuring WLP.

WAINST015E The following command failed:

C:\WA\BKM95\appservertools\startAppServer.bat -directclean

WAINST035I For more details see the installation log file: C:\WA\BKM95\logs\serverinst_9.5.0.04.log.

```

Cause and solution:

It might occur that the previous WebSphere Application Server Liberty process, named **javaw**, is still up and running and is already using the application ports.

To solve the problem, proceed as follows:

1. Check if there is a **javaw** process running which is related to the previous version 9.5 fix pack x instance, using the Java version installed in the `JavaExt9.5.0._OLD_FP` path, for example `JavaExt9.5.0.02\jre\jre\bin\javaw.exe`.
2. If you find the **javaw** process, stop it and restart the upgrade process.

Error received when creating MSSQL database

Error received when creating MSSQL database

When creating the database for MSSQL, you might receive an error similar to the following:

```
'CREATE SCHEMA' must be the first statement in a query batch.
```

Cause and solution

When you run the `configureDb` script specifying the `execsql=false` parameter, the `customSQL.sql` and `customSQLAdmin.sql` are created and stored locally.

Before sending them to the database administrator, perform the following steps:

1. Add the following to strings to the `customSQL.sql` file:

```

CREATE SCHEMA EVT
GO
CREATE SCHEMA PLN
GO
CREATE SCHEMA MDL
GO
CREATE SCHEMA LOG
GO
CREATE SCHEMA DWB
GO

```

2. Replace all semicolons (;) with the string `go` in the `customSQL.sql`.
3. Send both files to the database administrator.
4. The database administrator must run the `customSQLAdmin.sql` file on the database server.
5. The database administrator must run the `customSQL.sql` file on the new database created with the previous query.

For more information about the `execsql` parameter and the `configureDb` script, see [Database configuration - configureDb script on page 430](#).

Incorrect collation settings in PostgreSQL database

Job ordering and extract from folders might work incorrectly when using a PostgreSQL database

If you are using a PostgreSQL database, your data might perform erratically if the collation feature is not enabled. To check the collation feature and configure it if necessary, perform the following steps:

1. From PostgreSQL command line, run the following command to check whether the collation feature is enabled:

```
\l
```

2. Verify the output and check the **Collate** column:

List of databases					
Name	Owner	Encoding	Collate	Ctype	Access privileges
dwc	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	
fips1	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	
postgre	postgres	UTF8	C	en_US.UTF-8	
postgres	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	
template0	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	=c/postgres +
					postgres=CTc/postgres
template1	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	=c/postgres +
					postgres=CTc/postgres

If the **Collate** column contains a value different from C, the data is incorrect.

3. To resolve the problem, stop WebSphere Application Server Liberty, as described in the topic about starting and stopping the application server in *Administration Guide*.
4. Run the following command to update the database and set the collation feature:

```
update pg_database set datcollate='C', datctype='ucs_baisc.UTF-8' where datname='TWS';
```

5. Run the following command to re-index the database:

```
REINDEX database "TWS";
```

6. Start WebSphere Application Server Liberty, as described in the topic about starting and stopping the application server in *Administration Guide*.

The collation feature is now enabled and the database works correctly. If you were performing an upgrade, you can return to the upgrade procedure:

- [Performing a direct upgrade from v 9.5.0.x or v 10.x.x to v 10.2.5 on page 238](#)
- [Parallel upgrade from version 9.5.0.x or 10.x.x to version 10.2.5 on page 257](#)
- [Parallel upgrade from version 9.4.0.x to version 10.2.5 on page 311](#)

Chapter 29. Uninstalling HCL Workload Automation manually

Steps to take when manually uninstalling the HCL Workload Automation master domain manager.

How to manually remove the HCL Workload Automation master domain manager.

Run the steps listed in the following topics to manually uninstall an HCL Workload Automation instance:

- [Uninstalling manually on Windows operating systems on page 410](#)
- [Uninstalling manually on UNIX operating systems on page 412](#)

Read the following topic to learn about known workaround for problems that might affect the HCL Workload Automation uninstall:

- [Problems during manual uninstall on page 414](#)

Uninstalling manually on Windows™ operating systems

Steps to take when manually uninstalling the HCL Workload Automation master domain manager on a Windows™ operating systems.

Run the following steps to manually remove an HCL Workload Automation master domain manager.



Note: If your RDBMS is based on Oracle, browse to the `TWA_home\usr\servers\engineServer\configDropins\overrides` path and check in the `datasource.xml` configuration file the net service name used for your database before uninstalling the master domain manager.

1. Shut down all HCL Workload Automation operations and processes

1. On a system prompt, go to the HCL Workload Automation installation path.
2. Set the environment by running the `twa_env.cmd` command.
3. Stop the dynamic agent by running the `ShutDownLwa` command.
4. Stop **netman**, **conman** and their child processes by running the `conman "shutdown` command.
5. Stop the event process by running the `conman stopmon` command.
6. Stop the application server process by running the `conman stopappservman` command.
7. In the task manager, verify that the following processes are inactive:

```
netman
appservman
java
mailman
monman
```

As an alternative, you can also stop all processes by shutting down the related HCL Workload Automation services from the services panel.

2. Delete the HCL Workload Automation services

If you are uninstalling the master domain manager, you must delete the following services:

```
twc_tokensrv_TWS_user
twc_maestro_TWS_user
twc_ssm_agent_TWS_user
twc_netman_TWS_user
twc_cpa_agent_TWS_user
IBMWASService - TWS_user
```

The command to delete a service is:

```
sc delete service_name
```

When you finished, check that the following services are no longer listed in the active services for the *TWS_user*.

Workload Scheduler

Netman

Token service

Common Platform agent

If any of these services is still in the list, reboot the system and check again.

3. Delete the HCL Workload Automation files

Delete all the files under the *TWA_install_dir* directory.

4. Drop the HCL Workload Automation tables to the RDBMS

On DB2:

Run the following steps:

1. From the program menu, open the DB2 command line processor (CLP).
2. Look for the database name by running the command:

```
list db directory
```

3. If you see an entry named *your_db_name* associated to the HCL Workload Automation instance, run the command:

```
drop db your_db_name
```

If the master domain manager was installed on the DB2 client, run steps 1 and 5 also on the system where the master domain manager is installed.

On ORACLE:

Run the following steps:

1. Access the ORACLE command line.
2. Run the command:

```
sqlplus system/password@net_service_name
```

3. Delete all the tables related to the HCL Workload Automation instance by running the command:

```
drop user ORACLE_TWS_user cascade;
```

Uninstalling manually on UNIX™ operating systems

Steps to take when uninstalling HCL Workload Automation master domain manager manually on UNIX™ operating systems.

To manually remove an HCL Workload Automation master domain manager complete the following steps.



Note: If your RDBMS is based on Oracle, browse to the `TWA_DATA_DIR/usr/servers/engineServer/configDropins/overrides` path and check in the `datasource.xml` configuration file the net service name used for your database before uninstalling the master domain manager.

1. Shut down all HCL Workload Automation operations and processes

1. On a system prompt, go to the HCL Workload Automation installation path.
2. Set the environment by running the `twc_env.sh` command.
3. Stop the dynamic agent by running the `ShutDownLwa` command.
4. Stop the event processor by running the `conman stopmon` command.
5. Stop the application server process by running the `conman stopappservman` command.
6. Stop **netman**, **conman**, and their child processes by running the `conman "shut;wait"` command.
7. To verify that the following processes are inactive, run the command `ps -ef | grep process_name`.

```
netman
appservman
java
mailman
monman
```

2. Delete the HCL Workload Automation files

Delete all the files under the `TWS_install_dir` directory.



Note: The `TWS_install_dir` directory is not the HCL Workload Automation directory, as that might also contain a Dynamic Workload Console installation.

3. Drop the HCL Workload Automation tables into the RDBMS

On DB2:

Complete the following steps:

1. From the program menu, open the DB2 command-line processor (CLP)
2. Look for the database name by running the command:

```
list db directory
```


3. If you see an entry named `your_db_name` associated to the HCL Workload Automation instance, run the command:

```
drop db your_db_name
```

4. If you see an entry named `your_db_name` associated to the HCL Workload Automation instance, run the command:

```
uncatalog db your_db_name_DB
```

5. To see which node is attached to the master domain manager, run the command:

```
list node directory
```

6. Run the command:

```
uncatalog node your_node
```

If the master domain manager was installed on the DB2 client, perform the same procedure also on the workstation where the master domain manager is installed.

On ORACLE:

Complete the following steps:

1. Access the Oracle command line.
2. Run the command:

```
sqlplus system/password@net_service_name
```

3. Delete all the tables related to the HCL Workload Automation instance by running the command:

```
drop user ORACLE_TWS_user cascade;
```

4. Delete the HCL Workload Automation administrative user that was created at installation time.

5. Delete the HCL Workload Automation registries

1. Edit the `/etc/TWS/TWSRegistry.dat` file.
2. Delete the lines tagged with **TWS_user**.
3. Go to the `/etc/TWA` directory which contains two files for each HCL Workload Automation instance installed.
4. Look for the properties file that applies to the HCL Workload Automation instance to remove.
5. Delete the properties file and the file with the same filename and extension `.ext`.
6. Delete the `/etc/init.d/tebet1-tws_cpa_agent_TWS_user` directory.

6. Remove the Common Platforms Agent configuration file

Remove the file named `/etc/teb/teb_tws_cpa_agent_TWS_user.ini`.

7. Remove WebSphere Application Server Liberty

Delete all files located in the `IWA_install_dir/wlp` directory and the `wlp` directory itself.



Note: Do not delete the above files and directories if other components are installed and using WebSphere Application Server Liberty, such as the Dynamic Workload Console.

Problems during manual uninstall

The following problem might occur during a manual uninstall:

- [File deletion on Windows too slow on page 414](#)

File deletion on Windows™ too slow

When manually deleting files during a manual uninstallation, the deletion of the files in the path `$TWA_DIR\TWS\stdlist\yyyy.mm.dd\Onnnn.hhmm` is unacceptably slow.

Cause and solution:

This problem is caused by a known Microsoft™ issue on Windows™ operating systems. It occurs when you try to delete the indicated files on the Windows™ system after having uninstalled the master domain manager. To prevent the problem from occurring use **Shift-Canc** to remove these files instead of using the **Delete** menu option, moving them to the recycle bin, or using the **Canc** key on the keyboard.

Part VIII. Uninstalling

An overview on how to uninstall the product.

Uninstalling the product does not remove files created after HCL Workload Automation was installed, nor files that are open at the time of uninstallation. If you do not need these files, you must remove them manually. If you intend to reinstall and therefore need to use the files, make a backup before starting the installation process. The uninstallation does not remove your DB2® or Oracle database.



Note: To manually uninstall HCL Workload Automation, see [Uninstalling HCL Workload Automation manually on page 410](#)

Chapter 30. Uninstalling the main components

Before you begin

Before performing the uninstallation, whether if the following conditions are met:

1. Ensure that the user running the process has the following authorization requirements:

Windows™ operating systems

If you set the Windows User Account Control (UAC), your login account must be a member of the Windows™ **Administrators** group or domain administrators with the right, **Act as Part of the Operating System**.

If you set the Windows User Account Control (UAC) on the workstation, you must run the installation as **administrator**.

UNIX™ and Linux™ operating systems

If the component was installed with root privileges, **root** access is required. If you performed a **no-root installation**, specify the same user used for installing the component.

2. Ensure that all HCL Workload Automation processes, services and the WebSphere Application Server Liberty process are stopped, and that there are no active or pending jobs. For information about stopping the processes and services see the topic about starting and stopping processes on a workstation in the *User's Guide and Reference*.

About this task

The following section describes how to uninstall the following components:

- master domain manager or its backup
- dynamic domain manager or its backup
- agents

Results

The uninstallation removes the product files, the registry keys, and on Windows operating systems, also the services. It also removes the binaries related to the installed HCL Workload Automation agent.

The uninstallation program does not remove the HCL Workload Automation configuration files.

Uninstalling a backup master domain manager

About this task

To uninstall a backup master domain manager, perform the following steps:

1. To uninstall the backup master domain manager, you must first remove it from the plan. Set the workstation running the backup master domain manager to `ignore`, using either the `composer mod cpu workstation_name>` command or from the Dynamic Workload Console.
2. Run JnextPlan to generate the new production plan so that the backup master domain manager is removed from the plan.
3. Run the uninstall script.

- a. Change directory using the following command:

```
cd TWA_home>/TWS/tws_tools
```

- b. Run the uninstallation process by running the script as follows:

Windows™ operating systems

```
cscript uninstall.vbs --prompt no --wauser user_name>
```

UNIX™ and Linux™ operating systems

```
./uninstall.sh --prompt no --wauser user_name
```

where, `user_name>` represents the user for which you want to uninstall the backup master domain manager. The procedure runs without prompting the user to confirm the uninstallation.

4. Run JnextPlan to update the plan with the changes.

Uninstalling a master domain manager

About this task

To uninstall a master domain manager, perform the following steps:

1. Run the uninstall script.
 - a. Change directory using the following command:

```
cd TWS_home/TWS/tws_tools
```

- b. Start the uninstallation process by running the script as follows:

Windows™ operating systems

```
cscript uninstall.vbs --prompt no --wauser user_name
```

UNIX® and Linux® operating systems

```
./uninstall.sh --prompt no --wauser user_name
```

where, `user_name` represents the user for which you want to uninstall the master domain manager. The procedure runs without prompting the user to confirm the uninstallation.

2. Drop the HCL Workload Automation tables to the RDBMS.

On DB2®:

Run the following steps:

- a. From the program menu, open the DB2® command-line processor (CLP).
- b. Look for the database name by running the command:

```
list db directory
```

- c. If you see an entry named `your_db_name` associated to the HCL Workload Automation instance, run the command:

```
drop db your_db_name
```

- d. If you see an entry named `your_db_name_DB` associated to the HCL Workload Automation instance, run the command:

```
uncatalog db your_db_name_DB
```

- e. To see which node is attached to the master domain manager system run the command:

```
list node directory
```

- f. Run the command:

```
uncatalog node your_node
```

If the master domain manager was installed on the DB2® client, run the same on the system where the master domain manager is installed.

On ORACLE:

Run the following steps:

- a. Access the ORACLE command line.
- b. Run the command:

```
sqlplus system/password@net_service_name
```

- c. Delete all the tables related to the HCL Workload Automation instance by running the command:

```
drop user ORACLE_TWS_user cascade;
```

3. Delete the HCL Workload Automation administrative user that was created at install time.

Results

The log files generated from this command are located in the following path:

On Windows operating systems

`TWA_home\logs`

On UNIX operating systems

`TWA_DATA_DIR/installation/logs`

Uninstalling the Dynamic Workload Console

Before you begin

Ensure that all HCL Workload Automation processes, services and the WebSphere Application Server Liberty process are stopped, and that there are no active or pending jobs. For information about stopping the processes and services see

the topic about starting and stopping processes on a workstation in the *HCL Workload Automation: User's Guide and Reference*.

About this task

To uninstall the Dynamic Workload Console, perform the following steps:

1. Change directory to the folder containing the uninstallation script:

```
cd DWC_INST_DIR/tools
```

2. Run the uninstallation process by running the script as follows:

Windows™ operating systems

```
cscript uninstall.vbs --prompt no
```

UNIX™ and Linux™ operating systems

```
./uninstall.sh --prompt no
```

The procedure runs without prompting the user to confirm the uninstallation.

Results

The log file generated by this command are located in:

On Windows operating systems

```
<DWC_home>\logs
```

On UNIX operating systems

```
<DWC_DATA_dir>/installation/logs
```

Uninstalling a dynamic domain manager or its backup

Authorization requirements to verify before uninstalling.

Before you begin

1. Ensure that all HCL Workload Automation processes, services and the WebSphere Application Server Liberty process are stopped, and that there are no active or pending jobs. For information about stopping the processes and services see *User's Guide and Reference*.
2. To maintain a correct hierarchy of the HCL Workload Automation network, see [Uninstalling a dynamic domain manager maintaining a correct hierarchy in the network on page 421](#).

About this task

To uninstall a dynamic domain manager or its backup, perform the following steps:

1. Run the uninstall script.

- a. Change directory using the following command:

```
cd <TWS_home>/TWS/tws_tools
```

- b. Start the uninstallation process by running the script as follows:

Windows™ operating systems

```
cscript uninstall.vbs --prompt no --wauser user_name>
```

UNIX® and Linux® operating systems

```
./uninstall.sh --prompt no --wauser user_name>
```

where, *user_name*> represents the user for which you want to uninstall the dynamic domain manager. The procedure runs without prompting the user to confirm the uninstallation.

2. Drop the HCL Workload Automation tables to the RDBMS.

On DB2®:

Run the following steps:

- a. From the program menu, open the DB2® command-line processor (CLP).
- b. Look for the database name by running the command:

```
list db directory
```

- c. If you see an entry named *your_db_name* associated to the HCL Workload Automation instance, run the command:

```
drop db your_db_name
```

- d. If you see an entry named *your_db_name_DB* associated to the HCL Workload Automation instance, run the command:

```
uncatalog db your_db_name_DB
```

- e. To see which node is attached to the dynamic domain manager system run the command:

```
list node directory
```

- f. Run the command:

```
uncatalog node your_node
```

If the dynamic domain manager was installed on the DB2® client, run the same on the system where the dynamic domain manager is installed.

On ORACLE:

Run the following steps:

- a. Access the ORACLE command line.
- b. Run the command:

```
sqlplus system/password@net_service_name
```


- c. Delete all the tables related to the HCL Workload Automation instance by running the command:

```
drop user ORACLE_TWS_user cascade;
```

3. Delete the HCL Workload Automation administrative user that was created at install time.

Uninstalling a dynamic domain manager maintaining a correct hierarchy in the network

To correctly uninstall a dynamic domain manager, perform the following steps:

1. Uninstall the dynamic agents connected to the dynamic domain manager you want to uninstall by using one of the procedures described in this section.
2. In the database, delete the definitions of the workstations of type AGENT that are connected to the dynamic domain manager that you are uninstalling. You can use either the Dynamic Workload Console workload designer or run the following command:

```
composer del ws agent_workstation_name
```

3. Delete the definitions of the workstations of type REM-ENG connected to the dynamic domain manager that you are uninstalling. You can use either the Dynamic Workload Console workload designer or run the following command:

```
composer del ws rem_eng_workstation_name
```

4. Delete the definitions of the workstations of type POOL connected to the dynamic domain manager that you are uninstalling. You can use either the Dynamic Workload Console workload designer or run the following command:

```
composer del ws pool_workstation_name
```

5. Delete the definitions of the workstations of type D-POOL connected to the dynamic domain manager that you are uninstalling. You can use either the Dynamic Workload Console workload designer or run the following command:

```
composer del ws dpool_workstation_name
```

6. Uninstall the dynamic domain manager.
7. Delete the definition of the workstations of type X-AGENT hosted by the dynamic domain manager that you are uninstalling. You can use either the Dynamic Workload Console workload designer, or run the following command:

```
composer del ws x-agent_workstation_name
```

8. Delete the definitions of the workstations of type BROKER of the dynamic domain manager that you are uninstalling. You can use either the Dynamic Workload Console workload designer or run the following command:

```
composer del ws broker_workstation_name
```

Uninstalling agents using the twsinst script

Before you begin

1. Before starting to uninstall, verify that the user running the uninstallation process has the following authorization requirements:

Windows™ operating systems

If you set the Windows User Account Control (UAC), your login account must be a member of the Windows™ **Administrators** group or domain administrators with the right, **Act as Part of the Operating System**.

If you set the Windows User Account Control (UAC) on the workstation, you must run the installation as **administrator**.

On UNIX™ and Linux™ operating systems:

To uninstall a fault-tolerant agent or a dynamic agent that was installed by the **root** user, the user must have **root** access.

To uninstall a fault-tolerant agent or a dynamic agent that was installed by a **non-root user**, the uninstaller must use the same login used to install the agent. To find the login value used at installation time for dynamic agents, see the read-only `InstallationLoginUser` parameter in the `JobManager.ini` configuration file on the agent.

2. Ensure that you have enough temporary space before starting the uninstallation process.
3. Ensure that all HCL Workload Automation processes and services are stopped, and that there are no active or pending jobs. For information about stopping the processes and services, see the topic about starting and stopping processes on a workstation.

Follow these steps to uninstall HCL Workload Automation agents using the twsinst script. Depending on the operating system, proceed as follows:

On Windows™ operating systems:

1. Ensure that all HCL Workload Automation processes and services are stopped, and that there are no active or pending jobs. For information about stopping the processes and services, see the topic about starting and stopping processes on a workstation.
2. Log on as administrator on the workstation where you want to uninstall the product.
3. **twsinst** for Windows™ is a Visual Basic Script (VBS) that you can run in CScript and WScript mode, from the `installation_dir\TWS`, run the twsinst script as follows:

```
cscript twsinst -uninst -uname username [-wait minutes]
[-lang lang_id]
[-work_dir working_dir]
```

The uninstallation is performed in the language of the locale and not the language set during the installation phase. If you want to uninstall agents in a language other than the locale of the computer, run the **twinsinst** script from the *installation_dir*\TWS as follows:

```
cscript twinsinst -uninst -uname user_name -lang language
```

where *language* is the language set during the uninstallation.

On UNIX™ and Linux™ operating systems:

1. Log on as root, or as the user who installed the agent, and change your directory to */installation_dir/TWS*.
2. From the TWS directory, run the twinsinst script as follows:

```
twinsinst -uninst -uname username [-wait minutes]
[-lang lang_id] [-work_dir working_dir]
```

The uninstallation is performed in the language of the locale and not the language set during the installation phase. If you want to uninstall agents in a language other than the locale of the computer, run the **twinsinst** script from the */installation_dir/TWS* as follows:

```
./twinsinst -uninst -uname user_name -lang language
```

where *language* is the language set during the uninstallation.

-uninst

Uninstalls the HCL Workload Automation agent.

-uname username

The name of the user for which the HCL Workload Automation agent is uninstalled. If you installed the agent as the **root user**, this user name is not to be confused with the user performing the uninstallation. If you installed the agent as a **user different from root**, specify the same user name you used at installation time. In this case, the user performing the uninstallation and the user for which the agent is uninstalled are the same.

-wait minutes

The number of minutes that the product waits for jobs that are running to complete before starting the uninstallation. If the jobs do not complete during this interval, the uninstallation stops and an error message is displayed. Valid values are integers or **-1** for the product to wait indefinitely. The default is **60** minutes.

-lang lang_id

The language in which the `twinsinst` messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used.



Note: The **-lang** option is not to be confused with the HCL Workload Automation supported language packs.

-work_dir working_dir

The temporary directory used for the HCL Workload Automation installation process files deployment.

On Windows™ operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. If you do not manually specify a path, the path is set to `%temp%\TWA\twsversion_number>`, where `%temp%` is the temporary directory of the operating system.

On UNIX™ and Linux™ operating systems:

The path cannot contain blanks. If you do not manually specify a path, the path is set to `/tmp/TWA/twsversion_number>`.

The following is an example of a `twsinst` script that uninstalls the HCL Workload Automation agent, originally installed for user named **twuser**:

On Windows™ operating systems:

```
cscript twsinst -uninst -uname TWS_user
```


On UNIX™ and Linux™ operating systems:

```
./twsinst -uninst -uname TWS_user
```

Uninstalling agents on IBM i systems

How to uninstall dynamic and z-centric agents on IBM i systems.

To uninstall the agents on an IBM i system by using the `twsinst` script, perform the following steps:

1. Ensure that all HCL Workload Automation processes and services are stopped, and that there are no active or pending jobs. For information about stopping the processes and services, see the section about starting and stopping Application server in *Administration Guide*.
2. Sign on as the user who performed the installation, either **QSECOFR** or an existing user with ALLOBJ authority. If you installed with a user different from **QSECOFR**, use the same user who performed the installation and specify the **allObjAuth** parameter to indicate that the user has the ALLOBJ authority. For more information about this parameter, see [Agent installation parameters on IBM i systems on page 159](#). You can find the name of the profile used to perform the installation in the `instUser` located in the `agent_data_dir/installation/instInfo`.
3.  **Note:** Only for dynamic agents, you have the option of installing using a user different from **QSECOFR** and with no specific authorizations. In this case, specify the same user who performed the installation.
4. Change your directory to `/installation_dir/TWS`. For example: `/home/user1/TWS` where `user1` is the name of HCL Workload Automation user.
5. From the `Installation directory\TWS` directory, run the `twsinst` script as follows:

```
twsinst -uninst -allObjAuth -uname username  
[-wait minutes] [-lang lang_id] [-work_dir working_dir]
```

-uninst

Uninstalls HCL Workload Automation.

uname *username*

The name of the user for which HCL Workload Automation is uninstalled. This user name is not the same as the user performing the installation.

-allObjAuth

If you are installing, upgrading, or uninstalling with a user different from the default **QSECOFR** user, this parameter specifies that the user has the required ALLOBJ authority. Ensure the user is existing and has ALLOBJ authority because the product does not verify that the correct authority is assigned. The same user must be specified when installing, upgrading or uninstalling the agent. If you are using the **QSECOFR** user, this parameter does not apply.

-uname *username*

The name of the user for which HCL Workload Automation is uninstalled.

If you are using the **QSECOFR** user or a user with **ALLOBJ authority**, this user name is not the same as the user performing the installation. If you are using a user **different from QSECOFR**, the user performing the installation and the user for which the agent is installed are the same.

-wait *minutes*

The number of minutes that the product waits for jobs that are running to complete before starting the uninstallation. If the jobs do not complete during this intervals the uninstallation stops and an error message is displayed. Valid values are integers or **-1** for the product to wait indefinitely. The default is **60** minutes.

-lang *lang_id*

The language in which the `twswinst` messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used.

-work_dir *working_dir*

The temporary directory used for the HCL Workload Automation installation process files deployment. If you do not manually specify a path, the path is set to `/tmp/TWA/twsversion_number>`.

The following example shows a `twswinst` script that uninstalls the HCL Workload Automation agent, originally installed for **twswuser** user:

On IBM i systems:

```
./twswinst -uninst -uname TWS_user -allObjAuth
```

The twswinst script log files on IBM i systems

About this task

The twswinst log file name is:

Where: `<TWS_INST_DIR>/twswinst_IBM_i_TWS_user^product_version.log`

TWS_INST_DIR

The HCL Workload Automation installation directory. The default installation directory is `/home/TWS_user`.

TWS_user

The name of the user for which HCL Workload Automation was installed, that you supplied during the installation process.

product_version

Represents the product version. For example, for version 10.2.5 of the product, the value is 10.2.5.00

Appendix A. Reference

Contains the detailed syntax and explanation for all parameters of the commands required for the command-line installation:

- [Optional password encryption - secure script on page 427](#)
- [Database configuration - configureDb script on page 430](#)
- [Server components installation - serverinst script on page 442](#)
- [Dynamic Workload Console installation - dwcinst script on page 456](#)
- [Agent installation parameters - twsinst script on page 119](#)
- [File Proxy installation - fileproxyinst script on page 478](#)
- [File Proxy start - fileproxystart script on page 480](#)
- [File Proxy stop - fileproxystop script on page 481](#)
- [File Proxy uninstallation - uninstall script on page 481](#)
- [Certificates download to dynamic agents and fault-tolerant agents - AgentCertificateDownloader script on page 482](#)

Optional password encryption - secure script

Optionally encrypt the passwords you use to install, upgrade, and manage HCL Workload Automation.

This section lists and describes the parameters of the secure script. The secure command uses the AES method and prints the encrypted password to the screen or saves it to a file. This command is available by default on all HCL Workload Automation components.



Note: Use this script only to encrypt passwords used during the installation and upgrade processes.

You can either:

- Define a custom passphrase by using the **passphrase** argument and defining the **SECUREWRAP_PASSPHRASE** environment variable in the same shell session in which you run the command using the encrypted password. Ensure you set the **SECUREWRAP_PASSPHRASE** environment variable to the same value as the **passphrase** argument. You can use encrypted passwords only in association with the specific passphrase used to encrypt them.
- Use the standard encryption method provided with the secure command. In this case, you simply specify the **password** parameter.



Note: It is important you understand the limits to the protection that this method provides. The custom passphrase you use to encrypt the passwords is stored in clear format in the `passphrase_variables.xml` file, stored



in `configureDropin`. To fully understand the implications of this method, it is recommended you read the information provided by Open Liberty at the link [Password encryption limitations](#).

Syntax

Windows operating systems:

```
secure -fips on|weak|off | -checksecurity | -updatesecurity | -securitystatus | {-password password
| -in file} [-fipscompliance true|false] [-des3toaes]
[[-passphrase passphrase] | [-useaeskeystore]] [-out file]
```

UNIX operating systems:

```
./secure -fips on|weak|off | -checksecurity | -updatesecurity | -securitystatus | {-password password
| -in file} [-fipscompliance true|false] [-des3toaes]
[[-passphrase passphrase] | [-useaeskeystore]] [-out file]
```

z/OS operating systems:

```
secure -fips on|weak|off | -checksecurity | -updatesecurity | -securitystatus | {-password password
| -in file} [-fipscompliance true|false] [-des3toaes]
[[-passphrase passphrase] | [-useaeskeystore]] [-out file]
```

Arguments

-fips

Specifies your FIPS settings:

on

Select `on` to enable FIPS in **full** mode. In this mode, FIPS enforces the most rigorous cryptographic levels defined by the FIPS 140-3 standard.

weak

Select `weak` to enable FIPS in **weak** mode. In this mode, FIPS enforces the most rigorous cryptographic levels defined by the FIPS 140-3 standard, but supports also the SHA-1 and 3DES algorithms.

off

Select `off` to disable FIPS. In this mode, FIPS standards are not enforced, but the product is still robust and secure.

-checksecurity

Checks the encryption level for password encryption. If user passwords are encrypted with the AES algorithm, the command modifies the `useAESEncryptionAlgorithm` optman option to `yes`. If the encryption algorithm is different from AES, the command displays a warning message.

-updatesecurity

Changes the encryption algorithm for user passwords from 3DES to AES. To prevent communication problems, this change requires that all components in your environment are at version 10.2.5 or later.

The command checks and modifies the **useAESEncryptionAlgorithm** option, based on the encryption algorithm used in your environment for user passwords, as follows:

users with passwords in 3DES

the **useAESEncryptionAlgorithm** option is set to `no`.

users with passwords in AES

the **useAESEncryptionAlgorithm** option is set to `yes`.

For more information about optman options, see the topic about global options in *Administration Guide*.

-securitystatus

Displays security settings in your environment, for example the current FIPS mode.

-password

Specifies the password to be encrypted. This parameter is mutually exclusive with the **-in** parameter.

-in

Specifies the name and path of the file where you have stored the password to be encrypted. This parameter is mutually exclusive with the **-password** parameter.

-fipscompliance

Allows overriding the product's FIPS mode. For instance, if a master domain manager, agent, or Dynamic Workload Console has FIPS enabled and the option **-fipscompliance** `=false` is specified, FIPS is selectively disabled for the secure command. Conversely, passing **-fipscompliance** `=true` enforces FIPS for that command, regardless of the global setting.

-des3toaes

Converts the specified password from the Triple DES to the AES format.

-passphrase

Specifies the custom passphrase that is used to generate the key with which the command encrypts the password. If you set this parameter, inform the user who installs HCL Workload Automation that they must define the **SECUREWRAP_PASSPHRASE** environment variable in the same shell from which they run the installation command, and set it to the same value as the **passphrase** parameter. On Windows operating systems, the passphrase must be at least 8 characters long. This argument generates a password which can be reused for all HCL Workload Automation components. This parameter is mutually exclusive with the [-useaeskeystore on page 429](#) parameter, which generates a password which can be decrypted only on the local workstation and not reused for other components.

-useaeskeystore

Specifies that the secure command runs the encryption process using the AES keystore specified in the **encrypt keystore file** option and associated to the **encrypt label** alias. Both options are defined in the `localopts` file. The keystore is created automatically at installation time. Using this parameter ensures that passwords are encrypted with a unique key for each installation. Consequently, files encrypted on one component cannot be decrypted on another component due to differing encryption keys. For more information about the **encrypt**

keystore file option and the **encrypt label** alias, see the topic about localopts details in *Administration Guide*.

This parameter is mutually exclusive with the [-passphrase on page 429](#) parameter, which generates a password which can be reused for other components.

-base64 e

Specifies that the encoding process uses the **base64** format.

-out

Specifies the path and name of a file where the command stores the encrypted password. If you do not specify this parameter, the encrypted password is printed to the screen.

Examples

To encrypt password `MyPassword` with a strong passphrase, run the following command:

```
./secure -password MyPassword -passphrase de85pU!Mb5G2xewPgDva
```

To encrypt the password stored in file `MyFile` using the default passphrase and save the encrypted password to file `OutputFile`, run the following command:

```
secure -in C:\info\MyFile -out C:\info\OutputFile
```

Database configuration - configureDb script

This script creates and populates the HCL Workload Automation database

This script is typically used by the database administrator for creating and populating the HCL Workload Automation database. For a typical scenario, see [Creating and populating the database on page 59](#).

This section lists and describes the parameters that you can use to create and populate the HCL Workload Automation database.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

The log files generated from this command are located in the following path:

On Windows operating systems

`TWA_home\logs`

On UNIX operating systems

`TWA_DATA_DIR/installation/logs`

On z/OS operating system

`TWA_DATA_DIR/installation/logs`

Syntax for Windows operating systems

Show command usage

```
configureDb -? | --usage | --help
```

Retrieve the command parameters and values from a file

```
configureDb --propfile | -f [property_file]
```

General information

```
[--lang lang_id]
[--work_dir working_directory]
[--wlpdir wlp_directory]
[--componenttype MDM | DDM | DWC ]
[--dbadminuser db_admin_user]
--dbadminuserpw db_admin_password
--rdbms_type|-r DB2 | DB2Z | ORACLE | MSSQL | POSTGRESQL
[--dbname db_name]
[--dbuser db_user]
[--dbport db_port]
--dbhostname db_hostname
[--dbdriverpath db_driver_path]
--auth_type authentication_type ]
[--iwstname table_space_name]
[--iwstspath table_space_path]
[--iwslogtsname log_table_space]
[--iwslogtspath log_path_table_space]
[--iwsplantsname plan_table_space]
[--iwsplantspath plan_path_table_space]
[--execsql execute_sql]
```

Oracle-only configuration options

```
--dbpassword db_password
[--usePartitioning true | false ]
[--Usage_TsTempName IWS_temp_path]
[--skipdbcheck true | false]
```

Db2 for z/OS-only configuration options

```
[--zlocationname zOS_location_containing_db]
[--zbufferpoolname buffer_pool_in_zOS_location]
```

Syntax for UNIX operating systems

Show command usage

```
configureDb -? | --usage | --help
```

Retrieve the command parameters and values from a file

```
configureDb --propfile | -f [property_file]
```

General information

```

[--lang lang_id]
[--work_dir working_directory]
[--wlpdir wlp_directory]
[--componenttype MDM | DDM | DWC ]
[--dbadminuser db_admin_user]
--dbadminuserpw db_admin_password
--rdbmstype|-r DB2 | DB2Z | ORACLE | MSSQL | POSTGRESQL
[--dbname db_name]
[--dbuser db_user]
[--dbport db_port]
--dbhostname db_hostname
[--dbdriverpath db_driver_path]
[--iwstname table_space_name]
[--iwstspath table_space_path]
[--iwslogtsname log_table_space]
[--iwslogtspath log_path_table_space]
[--iwsplantsname plan_table_space]
[--iwsplantspath plan_path_table_space]
[--execsql execute_sql ]

```

Oracle-only configuration options

```

--dbpassword db_password
[--usePartitioning true | false ]
[--Usage_TsTempName IWS_temp_path]
[--skipdbcheck true | false]

```

Db2- and PostgreSQL-only security options

```

[--sslkeysfolder keystore_truststore_folder]
[--sslpassword ssl_password]
[--dbsslconnection true | false]

```

Db2 for z/OS-only configuration options

```

[--zlocationname zOS_location_containing_db]
[--zbufferpoolname buffer_pool_in_zOS_location]

```

Syntax for z/OS operating system**Show command usage**

```
configureDb -? | --usage | --help
```

Retrieve the command parameters and values from a file

```
configureDb --propfile | -f [properties_file]
```

General information

```

[--lang lang_id]
[--work_dir working_directory]
[--wlpdir wlp_directory]
[--dbadminuser db_admin_user]
[--componenttype DWC ]
--dbadminuserpw db_admin_password
--rdbmstype|-r DB2 | DB2Z | ORACLE | MSSQL | POSTGRESQL
[--dbname db_name]
[--dbuser db_user]
[--dbport db_port]
--dbhostname db_hostname
[--dbdriverpath db_driver_path]
[--iwstname table_space_name]
[--iwstspath table_space_path]
[--iwslogtsname log_table_space]
[--iwslogtspath log_path_table_space]
[--iwsplantsname plan_table_space]
[--iwsplantspath plan_path_table_space]
[--execsql execute_sql ]

```

Db2 for z/OS-only configuration options

```

[--zlocationname zOS_location_containing_db]
[--zbufferpoolname buffer_pool_in_zOS_location]

```

Database configuration parameters

-? | --usage | --help

Displays the command usage and exits.

--propfile|-f [properties_file]

Optionally specify a properties file containing custom values for `configureDb` parameters. The default file for the server components is `image_location/TWS/interp_name/configureDb.properties`, while the default file for the Dynamic Workload Console is `image_location/configureDb.properties`. Specifying a properties file is suggested if you have a high number of parameters which require custom values. You can also reuse the file with minimal modification for several installations. If you create a custom properties file, specify its name and path with the **-f** parameter.

--lang lang_id

The language in which the messages returned by the command are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used. If neither **--lang** nor LANG are used, the default codepage is set to SBCS. For a list of valid values for these variables, see the following table:

Table 25. Valid values for -lang and LANG

parameter

Language	Value
Brazilian Portuguese	pt_BR

Table 25. Valid values for -lang and LANG parameter (continued)

Language	Value
Chinese (traditional and simplified)	zh_CN, zh_TW
English	en
French	fr
German	de
Italian	it
Japanese	ja
Korean	ko
Russian	ru
Spanish	es



Note: This is the language in which the installation log is recorded and not the language of the installed component instance. The command installs all languages as default.

--work_dir

The working directory where you extract the installation image. It also contains the output produced by the command, such as the SQL statements if you set the **execsql** parameter to **false**. The default value is `/tmp` on UNIX operating systems and `C:\tmp` on Windows operating systems.

[--wlpdir wlp_directory]

The path to Open Liberty installation directory. Open Liberty is used to decrypt the passwords you provide in encrypted form. This parameter is required only if you encrypt your passwords with the **{xor}** or **{aes}** encoding.

--componenttype MDM | DDM | DWC

The HCL Workload Automation component for which the database is installed. This parameter is optional. Supported values are:

MDM

master domain manager. Applies only to distributed operating systems.

DDM

dynamic domain manager. Applies only to distributed operating systems.

DWC

Dynamic Workload Console (it comprises the Federator).

--dbadminuser *db_admin_user*

The database administrator user who creates the HCL Workload Automation or Dynamic Workload Console schema objects on the database server. This parameter is optional. Depending on the database vendor, the default values are as follows:

db2admin

when **--rdbmstype** is set to `DB2`

sysadm

when **--rdbmstype** is set to `DB2Z`

system

when **--rdbmstype** is set to `ORACLE`

sa

when **--rdbmstype** is set to `MSSQL`

--dbadminuserpw *db_admin_password*

The password for the DB administrator user who creates the HCL Workload Automation schema objects on the database server. This parameter is required. Special characters are not supported. You can optionally encrypt the password. For more information, see [Encrypting passwords \(optional\) on page 58](#).

--rdbmstype|-r *rdbms_type*

The database type. Supported databases are:

- **DB2**
- **ORACLE**
- **MSSQL** This value applies to MSSQL and supported MSSQL cloud-based databases.
- **POSTGRESQL**
-

This parameter is required and has no default value.

--dbname *db_name*

The name of the HCL Workload Automation or Dynamic Workload Console database. This parameter is optional and case sensitive. Depending on the component that you are installing and the database vendor, the default values are as follows:

When installing the server components**TWS**

when **--rdbmstype** is set to `DB2`

orcl

when **--rdbmstype** is set to `ORACLE`

TWS

when **--rdbmstype** is set to `MSSQL`

null

when **--rdbmstype** is set to `DB2z`

TWS

when **--rdbmstype** is set to `POSTGRESQL`

When installing the Dynamic Workload Console

This parameter is optional and case sensitive. Depending on the component that you are installing and the database vendor, the default values are as follows:

TDWC

when **--rdbmstype** is set to `DB2`

TDWC

when **--rdbmstype** is set to `DB2Z`

orcl

when **--rdbmstype** is set to `ORACLE`

TDWC

when **--rdbmstype** is set to `MSSQL`

TDWC

when **--rdbmstype** is set to `POSTGRESQL`

--dbuser *db_user*

The database user that has been granted access to the HCL Workload Automation or Dynamic Workload Console tables on the database server. This parameter is optional. Depending on the component that you are installing and the database vendor, the default values are as follows:

When installing the server components

db2tws

when **--rdbmstype** is set to `DB2`

twSORA

when **--rdbmstype** is set to `ORACLE`

sa

when **--rdbmstype** is set to `MSSQL`

null

when **--rdbmstype** is set to `DB2Z`

postgres

when **--rdbmstype** is set to `POSTGRESQL`

When installing the Dynamic Workload Console**db2dwc**

when **--rdbmstype** is set to `DB2`

root

when **--rdbmstype** is set to `DB2Z`

twosora

when **--rdbmstype** is set to `ORACLE`

sa

when **--rdbmstype** is set to `MSSQL`

--dbport *db_port*

The port of the database server. This parameter is optional. Depending on the database vendor, the default values are as follows:

50000

when **--rdbmstype** is set to `DB2`

446

when **--rdbmstype** is set to `DB2Z`

1521

when **--rdbmstype** is set to `ORACLE`

1433

when **--rdbmstype** is set to `MSSQL`

5432

when **--rdbmstype** is set to `POSTGRESQL`

--dbhostname *db_hostname*

The host name or IP address of database server. This parameter is required. If you are configuring the database in SSL mode, ensure you specify the fully qualified hostname used in the database client CN. This value is the same as the fully qualified hostname of the workstation where the database is installed.

--dbdriverpath *db_driver_path*

The path where the database drivers are stored. This parameter is optional, but becomes required when you set the **rdbmstype** parameter to **DB2Z**. If you use Db2 for z/OS with the Dynamic Workload Console version 10.2.4 or later, transfer the drivers in binary mode from the directory where you installed Db2 for z/OS to a directory of your choice. Specify the driver path using this parameter.

By default, the configuration script references the JDBC drivers supplied with the product images. If your database server is not compatible with the supplied drivers, then contact your database administrator for the correct version to use with your database server and specify the driver path using this parameter. Ensure you provide the same path in the configureDb, serverinst, and dwcinst commands.

--iwstsnam|-tn *table_space_name*

The name of the tablespace for HCL Workload Automation or Dynamic Workload Console data. This parameter is optional for all databases with the exception of the Oracle database. The default value for all databases other than Oracle is:

For all operating systems, except z/OS

TWS_DATA

For z/OS operating system

TWSDATA

--iwstspath|-tp *table_space*

The path of the tablespace for HCL Workload Automation or Dynamic Workload Console data. This parameter is optional. The default value for all databases other than Oracle is:

For all operating systems, except z/OS

TWS_DATA

For z/OS operating system

TWSDATA

Only on Windows systems hosting an MSSQL database, ensure the folder for the tablespace is already existing before running the configureDb command and specify the path using this parameter. Specify the path using forward slashes (/), for example: `c: /<my_path> /TWS_DATA`.

--iwslogtsnam|-ln *log_table_space*

The name of the tablespace for HCL Workload Automation log. This parameter is optional for all databases with the exception of the Oracle database. The default value for all databases other than Oracle is **TWS_LOG**. This parameter applies only to the server components.

--iwslogtspath|-lp *log_path_table_space*

The path of the tablespace for HCL Workload Automation log. This parameter is optional. The default value for all databases other than Oracle is **TWS_LOG**. This parameter applies only to the server components. Only on Windows systems hosting an MSSQL database, ensure the folder for the tablespace is already existing before running the configureDb command and specify the path using this parameter. Specify the path using forward slashes (/), for example: `c: /<my_path> /TWS_LOG`.

--iwsplantsnam|-pn *plan_table_space*

The name of the tablespace for HCL Workload Automation plan. This parameter is optional for all databases with the exception of the Oracle database. The default value for all databases other than Oracle is **TWS_PLAN**. This parameter applies only to the server components.

--iwsplantspath|-pp *plan_path_table_space*

The path of the tablespace for HCL Workload Automation plan. This parameter is optional. The default value for all databases other than Oracle is **TWS_PLAN**. This parameter applies only to the server components.

Only on Windows systems hosting an MSSQL database, ensure that the folder for the tablespace is already existing before running the configureDb command and specify the path using this parameter. Specify the path using forward slashes (/), for example: `c: / <my_path> /TWS_PLAN`.

--execsql|-es *execute_sql*

Set to **true** to generate and run the SQL file, set to **false** to generate the SQL statement without running it. The resulting files are stored in the path defined in the **--work_dir** parameter. This option is useful if you want to review the file before running it. This parameter is optional. The default value is **true**.

--auth_type

This parameter applies only to Windows operating systems. Specify the authentication type. Supported values are as follows:

SQLSERVER

Enables MSSQL authentication type. Only the user specified with the **--dbadminuser** parameter has the grants to administer the HCL Workload Automation or Dynamic Workload Console database.

WINDOWS

Enables Windows authentication type. The Windows user you used to log on to the workstation is assigned the grants to administer the HCL Workload Automation or Dynamic Workload Console database.

The default value is **SQLSERVER**.

Oracle-only configuration syntax**--dbpassword *db_password***

The password for the user that has been granted access to the HCL Workload Automation or Dynamic Workload Console tables on the database server. This parameter is required only if you are using an Oracle database. Special characters are not supported. You can optionally encrypt the password. For more information, see [Encrypting passwords \(optional\) on page 58](#).

--usePartitioning

Only applies when installing the master domain manager. Set to **true** if you want to use the Oracle partitioning feature, otherwise set it to **false**. This parameter is optional. The default value is **true**.

--Usage_TsTempName *IWS_temp_path*

Only applies when installing the master domain manager. The path of the tablespace for HCL Workload Automation temporary directory. This parameter is optional. The default value is **TEMP**.

--skipdbcheck

This parameter specifies whether the check on the existence of the Workload Automation schema for the Oracle user is performed or not. By default, the parameter is set to **false** and a check is performed on the Oracle user. If the user does not exist, the script then proceeds to create the user and the Workload Automation schema.

If you have already created your Oracle user, set this parameter to **true**. As a result, the check is skipped and the schema creation is performed also if the Oracle user is already existing.

This parameter is optional.

Db2- and PostgreSQL-only security options**--sslkeyfolder *keystore_truststore_folder***

The name and path of the folder containing certificates in PEM format. The installation program automatically processes the keystore and truststore files using the password you specify with the **--sslpassword** parameter. The folder must contain the following files:

- **ca.crt**

The Certificate Authority (CA) public certificate. Note that if certificates being installed are part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then this file must contain the Root CA certificate only. Any Intermediate CA certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.



Note: From V10.2.3, if certificates being installed are part of a chain, the `ca.crt` can contain also the intermediate CAs. In this case, it must begin with one or more intermediate CA certificates and end with the Root ca.

- **tls.key**

The private key of the end user certificate for the instance to be installed.

- **tls.crt**

The public part of the previous key, that is the end user certificate.

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

You can optionally create a subfolder to contain one or more `*.crt` files to be added to the server truststore as trusted CA, whose name must be `additionalCAs`. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or DB2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**. Note that if the end user certificate being installed in the instance is part of a chain consisting of 3 or more certificates (one Root CA, followed

by one or more Intermediate CAs, followed by the end user certificate), then the Intermediate CAs certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

For further information about how to generate custom certificates, see the topic about managing certificates using Certman in *HCL Workload Automation: Planning and Installation*.

If you are configuring the database in SSL mode, ensure you store in the `additionalCAs` folder the database CA certificate with extension `.crt`, for example, `db_ca.crt`.

This parameter is required if you set the **dbsslconnection** parameter to true.

--sslpassword *ssl_password*

The password for the certificates.

For more information, see [sslkeysfolder on page 450](#).

You can optionally encrypt the password using the secure script. For more information, see [Optional password encryption - secure script on page 427](#).

This parameter is required if you set the **dbsslconnection** parameter to true.

--dbsslconnection

Specify whether you want to enable SSL connection to the database. Supported values are `true` and `false`. The default value is `false`. If you set this parameter to `true`, the **sslkeysfolder** and **sslpassword** parameters become mandatory. Ensure the database client certificate specifies, as its Common Name (CN), the fully qualified hostname of the workstation where the database is installed.

Db2 for z/OS-only configuration syntax

--zlocationname *zos_location_containing_db*

The name of an already existing location in the z/OS environment that will contain the new database. The default value is **LOC1**.

--zbufferpoolname *buffer_pool_in_zos_location*

The name of an already existing buffer pool created in the location specified by `--zlocationname`. The default value is BP32K.

Comments



Note: The following parameters are also required when installing the master components and their values must be the same:

- **--rdbmstype**
- **--dbhostname**
- **--dbport**



- **--dbname**
- **--dbuser**

Server components installation - serverinst script

The master domain manager, backup domain manager, dynamic domain manager, backup dynamic domain manager, and installation parameters that can be defined for the serverinst script.

This section lists and describes the parameters that are used when running a serverinst script to install the master domain manager and backup domain manager, dynamic domain manager, and backup dynamic domain manager.

Certificates are now required when installing or upgrading HCL Workload Automation. You can no longer install nor upgrade HCL Workload Automation without securing your environment with certificates. The required certificates are:

- ca.crt
- tls.key
- tls.crt

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

The log files generated from this command are located in the following path:

On Windows operating systems

TWA_home\logs

On UNIX operating systems

TWA_DATA_DIR/installation/logs

Syntax

On Windows™ operating systems:

Show command usage

```
cscript serverinst.vbs -? | --usage | --help
```

Retrieve the command parameters and values from a properties file

```
cscript serverinst.vbs --propfile|-f [properties_file]
```

General information

```
cscript serverinst.vbs
--acceptlicense yes|no
[--lang lang_id]
```

```
[--inst_dir install_dir]
[--work_dir working_dir]
[--skipcheckprereq true|false]
[--skipcheckemptydir true|false]
[--skipusercheck true|false]
```

Configuration information for the data source

```
--rdbmstype|-r DB2 | ORACLE | MSSQL | POSTGRESQL
[--dbname db_name]
[--dbuser db_user]
--dbpassword db_password
[--dbport db_port]
--dbhostname db_hostname
[--dbdriverpath db_driver_path]
```

Licensing information

```
--licenseserverid license_server_ID
[--licenseserverurl license_server_URL]
[--licenseproxyserver license_proxy_server_address]
[--licenseproxyport license_proxy_server_port]
[--licenseproxyuser license_proxy_server_user]
[--licenseproxypassword license_proxy_server_password]
```

Security options

```
--sslkeysfolder keystore_truststore_folder
--sslpassword ssl_password
[--enablefips true | false]
```

User information

```
[--wadomain]
[--wauser wa_user]
[--wapassword wa_password]
```

Configuration information for the application server

```
--wlpdir|-w wlp_directory
[--httpsport https_port]
[--bootstrappport bootstrap_port]
[--bootsecport bootstrap_sec_port]
[--startserver true | false]
```

Configuration information for dynamic scheduling

```
[--displayname agent_name]
[--jimport port_number]
```

Configuration information for the master domain manager

```
[--componenttype MDM | DDM]
```

Configuration options when --componenttype is MDM

```
[--company company_name]
[--hostname hostname]
[--thiscpu workstation]

[--eifport eif_port]
[--brwksname broker_workstation_name]
[--brnetmanport broker_netman_port]
[--netmanport netman_port_number]
[--netmansslport netman_port_number]
```

Configuration options when --componenttype is DDM

```
[--domain domain_name]
--master mdm-domain_name
--mdmhttpsport mdm_https_port
--mdmbrokerhostnamemdm_hostname
[--eifport eif_port]
[--brwksname broker_workstation_name]
[--brnetmanport broker_netman_port]
[--netmanport netman_port_number]
[--netmansslport netman_port_number]
[--isforzos yes|no]
```

HCL Workload Automation encryption options

```
[--useencryption true | false]
[--encryptionpassword default]
```

On UNIX® operating systems

Show command usage

```
./serverinst.sh -? | --usage | --help
```

Retrieve the command parameters and values from a properties file

```
./serverinst.sh --propfile|-f [properties_file]
```

General information

```
./serverinst.sh
--acceptlicense yes|no
[--lang lang_id]
[--inst_dir install_dir]
[--work_dir working_dir]
[--data_dir wa_datadir]
[--skipcheckprereq true|false]
[--skipcheckemptydir true|false]
```


Configuration information for the data source

```

--rdbmstype|-r DB2 | ORACLE | MSSQL | POSTGRESQL
[--dbname db_name]
[--dbuser db_user]
--dbpassword db_password
[--dbport db_port]
--dbhostname db_hostname
[--dbdriverpath db_driver_path]
[--dbsslconnection true | false]

```

Licensing information

```

--licenseserverid license_server_ID
[--licenseserverurl license_server_URL]
[--licenseproxyserver license_proxy_server_address]
[--licenseproxyport license_proxy_server_port]
[--licenseproxyuser license_proxy_server_user]
[--licenseproxypassword license_proxy_server_password]

```

Security options

```

--sslkeysfolder keystore_truststore_folder
--sslpassword ssl_password
[--enablefips true | false]

```

User information

```

[--wauser wa_user]
[--wapassword wa_password]

```

Configuration information for the application server

```

--wlpdir|-w wlp_directory
[--httpsport https_port]
[--bootstrappport bootstrap_port]
[--bootsecport bootstrap_sec_port]
[--startserver true | false]

```

Configuration information for dynamic scheduling

```

[--displayname agent_name]
[--jimport port_number]

```

Configuration information for the master domain manager

```

[--componenttype MDM | DDM]

```

Configuration options when --componenttype is MDM

```
[--company company_name]
[--hostname hostname]
[--thiscpu workstation]

[--eifport eif_port]
[--brwksname broker_workstation_name]
[--brnetmanport broker_netman_port]
[--netmanport netman_port_number]
[--netmansslport netman_port_number]
```

Configuration options when --componenttype is DDM

```
[--domain domain_name]
--master mdm_domain_name
--mdmhttpsport mdm_https_port
--mdmbrokerhostname mdm_hostname
--eifport eif_port]
[--brwksname broker_workstation_name]
[--brnetmanport broker_netman_port]
[--netmanport netman_port_number]
[--netmansslport netman_port_number]
[--isforzos yes/no]
```

HCL Workload Automation encryption options

```
[--useencryption true | false]
[--encryptionpassword default]
```

Arguments

? | --usage | --help

Displays the command usage and exits.

--propfile|-f [properties_file]

Optionally specify a properties file containing custom values for serverinst parameters. The default file is

On Windows™ systems

```
image_dir>\TWS95_WIN_X86_64_SERVER\TWS\WINDOWS_X86_64\serverinst.properties
```

On UNIX® systems

```
image_dir>/TWS/interp>/serverinst.properties
```

Specifying a properties file is suggested if you have a high number of parameters which require custom values. You can also reuse the file with minimal modification for several installations. If you create a custom properties file, specify its name and path with the **-f** parameter.

General information

--acceptlicense yes/no

Specify whether to accept the License Agreement.

--lang *lang_id*

The language in which the messages returned by the command are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used. If neither **--lang** nor LANG are used, the default codepage is set to SBCS. For a list of valid values for these variables, see the following table:

Table 26. Valid values for -lang and LANG**parameter**

Language	Value
Brazilian Portuguese	pt_BR
Chinese (traditional and simplified)	zh_CN, zh_TW
English	en
French	fr
German	de
Italian	it
Japanese	ja
Korean	ko
Russian	ru
Spanish	es



Note: This is the language in which the installation log is recorded and not the language of the installed component instance. The command installs all languages as default.

--inst_dir *installation_dir*

The directory of the HCL Workload Automation installation. This parameter is optional. The default value is:

On Windows™ operating systems

```
C:\Program Files\wa
```

On UNIX® operating systems

```
/opt/wa
```

--work_dir *working_dir*

The temporary directory used by the program to deploy the installation process files. This parameter is optional. The default value is:

On Windows™ operating systems

```
C:\TMP
```

On UNIX® operating systems

```
/tmp/waversion_number
```

This parameter can also function as a backup directory during product upgrade with path `WORKING_DIR/backup`.

--data_dir wa_datadir

UNIX operating systems only. Specify the path to a directory where you want to store the logs and configuration files produced by HCL Workload Automation. This parameter is optional. If you do not specify this parameter, all data files generated by HCL Workload Automation are stored in the `TWA_home/TWSDATA` directory. This path is called, in the publications, `TWA_DATA_DIR`.

--skipcheckprereq true/false

If you set this parameter to `true`, HCL Workload Automation does not scan system prerequisites before starting the installation. This parameter is optional. The default value is `false`. For more information about the prerequisite check, see [Scanning system prerequisites for HCL Workload Automation on page 48](#).

--skipcheckemptydir true/false

Set this parameter to `true` to avoid checking whether the installation directory is empty. By default, this parameter is `false`, because starting from version 9.5 the installation directory must be empty. If you set this parameter to `true` and the installation directory is not empty, the installation process might fail.

--skipusercheck true/false

If you set this parameter to `true`, HCL Workload Automation, performs no checks on the user. This parameter is optional. The default value is `false`. By default, the following checks are performed:

local user

The script checks if the specified user is existing, has the correct access rights, and the password specified with the `wapassword` parameter is correct. If the user does not exist, the script creates it and grants it the correct access rights. If the specified password is incorrect, the script returns an error and the installation process stops.

domain user

The script checks if the specified user is existing, has the correct access rights, and the password specified with the `wapassword` parameter is correct. If the user does not exist, the script cannot create it and the installation process ends in error. If the user exists but does not have the correct access rights, the script assigns it the required rights. If the specified password is incorrect, the script returns an error and the installation process stops.

Configuration information for the data source

The values for these parameters must match the values defined by the database administrator when creating the database. For more information, see [Creating and populating the database on page 59](#) and browse to the topic for the database you are using.

--rdbms_type|-r *rdbms_type*

The database type. Supported databases are:

- **DB2**
- **ORACLE** This value applies to Oracle and Amazon RDS for Oracle
- **MSSQL** This value applies to MSSQL and MSSQL cloud-based databases.
- **POSTGRESQL**

This parameter is required and has no default value.

--dbname *db_name*

The name of the HCL Workload Automation database. This parameter is optional. The default value is **TWS**.

--dbuser *db_user*

The user that has been granted access to the HCL Workload Automation tables on the database server. This parameter is optional. The default value is **db2tws**.

--dbpassword *db_password*

The password for the user that has been granted access to the HCL Workload Automation or Dynamic Workload Console tables on the database server. This parameter is required. The default value is **password**. Special characters are not supported. You can optionally encrypt the password. For more information, see [Encrypting passwords \(optional\) on page 58](#).

--dbport *db_port*

The port of the database server. This parameter is optional. The default value is **50000**.

--dbhostname *db_hostname*

The host name or IP address of database server. This parameter is required.

--dbdriverpath *db_driver_path*

The path where the database drivers are stored. This parameter is optional. By default, the configuration script references the JDBC drivers supplied with the product images. If your database server is not compatible with the supplied drivers, then contact your database administrator for the correct version to use with your database server and specify the driver path using this parameter. Ensure you provide the same path in the `configureDb`, `serverinst`, and `dwcinst` commands.

--dbsslconnection *true / false*

Enables or disables the SSL connection to the database. The default value is **false**. This parameter applies only to DB2.

Licensing information**--licenseserverid *license_server_ID***

The ID of the license server which processes license usage information. This parameter is required. Instructions about how to obtain the ID of the license server which processes license usage information

are provided with the mail confirming your license. For more information, see the section about License computation model in *Administration Guide* and Enabling product license management in *HCL Workload Automation: Planning and Installation*.

--licenseserverurl

The URL of the license server which processes license usage information. This parameter is optional. The URL value is <https://api.hcltechsw.com/>.

--licenseproxyserver *license_proxy_server_address*

The IP of the proxy server which HCL Workload Automation, is expected to contact. This option is required if you are using a proxy server. The default value is null because it must be specified by the user.

--licenseproxyport *license_proxy_server_port*

The port of the proxy server the master domain manager uses to connect to the Internet. This option is required if you are using a proxy server. The default value is null because it must be specified by the user.

--licenseproxyuser *license_proxy_server_user*

The user of the proxy server the master domain manager uses to connect to the Internet. This option is required if you are using a proxy server protected by a user name and password. The default value is null because it must be specified by the user.

--licenseproxypassword *license_proxy_server_password*

The password of the proxy server which HCL Workload Automation, is expected to contact. This option is required if you are using a proxy server protected by a user name and password. The default value is null because it must be specified by the user. You can optionally encrypt the password using the secure script. For more information, see [Optional password encryption - secure script on page 427](#).

SSL configuration options

--sslkeyfolder *keystore_truststore_folder*

The name and path of the folder containing certificates in PEM format. The installation program automatically processes the keystore and truststore files using the password you specify with the **--sslpassword** parameter. The folder must contain the following files:

- **ca.crt**

The Certificate Authority (CA) public certificate. Note that if certificates being installed are part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then this file must contain the Root CA certificate only. Any Intermediate CA certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.



Note: From V10.2.3, if certificates being installed are part of a chain, the `ca.crt` can contain also the intermediate CAs. In this case, it must begin with one or more intermediate CA certificates and end with the Root ca.

- **tls.key**

The private key of the end user certificate for the instance to be installed.

- **tls.crt**

The public part of the previous key, that is the end user certificate.

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

You can optionally create a subfolder to contain one or more `*.crt` files to be added to the server truststore as trusted CA, whose name must be `additionalCAs`. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or DB2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**. Note that if the end user certificate being installed in the instance is part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then the Intermediate CAs certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

For further information about how to generate custom certificates, see the topic about managing certificates using Certman in *HCL Workload Automation: Planning and Installation*.

--sslpassword *ssl_password*

The password for the certificates.

For more information, see [sslkeysfolder on page 450](#).

You can optionally encrypt the password using the secure script. For more information, see [Optional password encryption - secure script on page 427](#).

--enablefips *false*

Specify whether you want to enable FIPS. The default value is `false`. This parameter is optional.

User information

--wauser *user_name*

The user for which you are installing HCL Workload Automation. This parameter is optional. The default value is the user performing the installation, unless you use a **user other than root**.

On UNIX operating systems, you can choose to install as the **root** user or as a **user other than root**. The following considerations apply:

- If the installer is the **root user**, the **wauser** parameter can be omitted if the *username* value is meant to be root, or can be set to a *username* value other than root.
- If the installer is **different from the root user**, consider the following points:
 - The **wauser** parameter can be omitted, but **wauser** is automatically set to the login name of the installer. If the installer specifies a **wauser** with a different *username* value, an error message is returned.
 - As a consequence, you can log in to the master domain manager uniquely with the user name of the installer.
 - The user must be enabled to login to the machine where the master domain manager is going to be installed.
 - Event Management triggers on files work only if the selected files are accessible to the user that was used for the installation.
 - Future upgrades, modifications, and removal of the master domain manager can be made exclusively with the same login used for installation.
 - When running **conman** and **composer** commands, it is mandatory to set the environment first, by using the `twc_env` script as described in [Setting the environment variables on page 206](#).

--wapassword *wauser_password*

The password for the user for which you are installing HCL Workload Automation.

On Windows operating systems

Supported characters for the password are alphanumeric, dash (-), underscore (_) characters, and `()!*~+.@!^`

On UNIX operating systems

Supported characters for the password are any alphanumeric, dash (-), underscore (_) characters, and `()!*~+.`

This parameter is required if you specify the **wauser** parameter. You can optionally encrypt the password using the `secure` script. For more information, see [Optional password encryption - secure script on page 427](#).

Configuration information for the application server

The values for these parameters must match the values defined when installing Open Liberty. For more information, see [Installing Open Liberty on page 56](#).

--wlpdir | *wlp_directory*

Open Liberty profile installation directory. This parameter is required.

--httpsport *https_port*

The HTTPS port. This parameter is optional. The default value is **31116**.

--startserver *true* | *false*

Specifies whether the Open Liberty server must be started after installation. This parameter is optional. The default value is **true**.

Configuration information for dynamic scheduling**--displayname *agent_name***

The name to be assigned to the agent. The name cannot start with a number. If the host name starts with a number, this parameter is required, otherwise it is optional. The default value is the host name of the workstation followed by `_1`.

--jimport *port_number*

The JobManager port number on which the dynamic domain manager is contacted by the dynamic agent. This parameter is optional. The default value is **31114**. The valid range is from 1 to 65535.

Configuration information for the master domain manager**--componenttype *MDM / DDM***

The workstation type being installed. Supported workstation types are:

MDM

master domain manager

DDM

dynamic domain manager

To install a backup domain manager, run the `serverinst` command on the workstation where you plan to install the backup domain manager. The `serverinst` command connects to the database you specify, discovers that a master domain manager is already installed, and proceeds to install a backup domain manager. The same procedure applies when installing a backup dynamic domain manager.

Configuration options when --componenttype is MDM**--company *company_name***

The name of the company. The company name cannot contain blank characters. The name is shown in program headers and reports. This parameter is optional. The default name is **COMPANY**.

--hostname *host_name*

The fully qualified host name or IP address on which the installation is performed. The default value is calculated at installation time.

--thiscpu *workstation*

The name of the HCL Workload Automation workstation for this installation. The name cannot exceed 16 characters, cannot start with a number, cannot contain spaces. If the host name starts with a number, this parameter is required, otherwise it is optional. This name is registered in the `localopts` file. The default name is the host name of the workstation.

--eifport *eif_port*

Specifies the Job Manager Event Integration Facility (EIF) port number. The default value is **31131**. The valid range is 1 to 65535.

--brwksname *broker_workstation_name*

The broker workstation name. This parameter is optional. The default value is the workstation host name followed by `_DWB`. It cannot start with a number.

--brnetmanport *port_number*

The TCP/IP port number used by the `netman` process to listen for communication from the dynamic domain manager. This parameter is optional. The default value is **41114**. The valid range is from 1 to 65535. This port number is registered in the `localopts` file. For each installation you must specify a different number. For more information about the `localopts` file, see the section about setting local options in *User's Guide and Reference*.

--netmanport *netman_port_number*

The TCP/IP port number used by the `netman` process to listen for communication from the master domain manager. This parameter is optional. The default value is **31111**. The valid range is from 1 to 65535. You can also set this parameter to `disabled`. In this case, you must provide a value for the `netmansslport` parameter, which enables SSL communication. This port number is registered in the `localopts` file, in the `nm port` attribute. For each installation you must specify a different number.

--netmansslport *SSL_port_number*

The TCP/IP port number used by the `netman` process to listen for communication from the master in SSL mode. The default value is 31113. The valid range is from 1 to 65535. You can also set the `netmansslport` parameter to `disabled` to use non-encrypted communication. If you set the `netmansslport` parameter to `disabled`, you must provide a value for the `netmanport` parameter. This port number is registered in the `localopts` file, in the `nm ssl full port` attribute. For each installation you must specify a different number.

Configuration options when --componenttype is DDM**--domain *domain_name***

Windows™ systems only. The domain name of the HCL Workload Automation user. This parameter is optional. The default value is **MASTERDM** when you install a master domain manager, and **DYNAMICDM** when you install a dynamic domain manager.

--master *mdm_domain_name*

The master domain manager name. It cannot start with a number. This parameter is required for the dynamic domain manager only. Do not specify when installing the master domain manager.

--mdmhttpsport *mdm_https_port*

The port of the master domain manager host used by the broker to contact master domain manager. This parameter is required. This parameter applies to the dynamic domain manager only. Do not specify when installing the master domain manager.

--mdmbrokerhostname *mdm_hostname*

The fully qualified host name or IP address of the master domain manager contacted by the dynamic domain manager. This parameter is required for the dynamic domain manager only. Do not specify when installing the master domain manager.

--eifport *eif_port*

Specifies the Job Manager Event Integration Facility (EIF) port number. The default value is **31131**. The valid range is 1 to 65535.

--brwksname *broker_workstation_name*

The broker workstation name. This parameter is optional. The default value is the workstation host name followed by **_DWB**. It cannot start with a number.

--brnetmanport *port_number*

The TCP/IP port number used by the `netman` process to listen for communication from the dynamic domain manager. This parameter is optional. The default value is **41114**. The valid range is from 1 to 65535. This port number is registered in the `localopts` file. For each installation you must specify a different number. For more information about the `localopts` file, see the section about setting local options in *User's Guide and Reference*.

--netmanport *netman_port_number*

The TCP/IP port number used by the `netman` process to listen for communication from the master domain manager. This parameter is optional. The default value is **31111**. The valid range is from 1 to 65535. You can also set this parameter to `disabled`. In this case, you must provide a value for the **netmansslport** parameter, which enables SSL communication. This port number is registered in the `localopts` file, in the **nm port** attribute. For each installation you must specify a different number.

--netmansslport *SSL_port_number*

The TCP/IP port number used by the `netman` process to listen for communication from the master in SSL mode. The default value is 31113. The valid range is from 1 to 65535. You can also set the **netmansslport** parameter to `disabled` to use non-encrypted communication. If you set the **netmansslport** parameter to `disabled`, you must provide a value for the **netmanport** parameter. This port number is registered in the `localopts` file, in the **nm ssl full port** attribute. For each installation you must specify a different number.

--isforzos *yes/no*

Set to **yes** if you want to connect the dynamic domain manager to only the Z controller. Set to **no** if you want to connect the dynamic domain manager to a master domain manager or, to both a master domain manager and a Z controller. This parameter is optional. The default value is **no**.

HCL Workload Automation encryption options**--useencryption *true / false***

Specifies whether HCL Workload Automation files must be encrypted at runtime. If you specify **true**, or do not set this parameter, files such as the Symphony file and the message queues are encrypted using AES-256 or AES-128 cryptography. By default, a fresh installation is automatically encrypted and the keystore password is *default*. To change the keystore password, use the **encryptionpassword** parameter. This parameter is optional.

--encryptionpassword *default*

The password for the keystore storing the AES-256 or AES-128 keys used to encrypt the files at runtime. This parameter is optional. The default value is *default*. You can optionally encrypt the password using the secure script. For more information, see [Optional password encryption - secure script on page 427](#).

Comments

Note: The values for the following parameters must match the values you provided when creating and populating the database:

- **--rdbmstype**
- **--dbhostname**
- **--dbport**
- **--dbname**
- **--dbuser**
- **--dbpassword**

Dynamic Workload Console installation - dwcinst script

This script installs the Dynamic Workload Console

This section lists and describes the parameters that are used when running a **dwcinst** script to install the Dynamic Workload Console. For a typical installation scenario, see [Installing the Dynamic Workload Console servers on page 110](#). If you are installing in a z/OS environment, see the topic about installing the Dynamic Workload Console in *HCL Workload Scheduler for Z: Planning and Installation*.

Certificates are now required when installing or upgrading HCL Workload Automation. You can no longer install nor upgrade HCL Workload Automation without securing your environment with certificates. The required certificates are:

- ca.crt
- tls.key
- tls.crt

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.



Note: Ensure that the **inst_dir** parameter is different from the directory of the installation image and it does not contain any HCL Workload Automation instances.

The log files generated from this command are located in the following path:

On Windows operating systems*DWC_home\logs***On UNIX operating systems***DWC_DATA_dir/installation/logs***On z/OS operating system***DWC_DATA_dir/installation/logs***Syntax for Windows operating systems****Show command usage**

```
dwcinst -? | --usage | --help
```

Retrieve the command parameters and values from a properties file

```
dwcinst --file | -f [properties_file]
```

General information

```

dwcinst
--acceptlicense yes|no
[--lang lang_id]
[--inst_dir install_dir]
[--skipcheckprereq true|false]
[--componenttype DWC | FED]

```

Configuration information for the data source

```

--rdbmstype|-r DB2 | DB2Z | ORACLE | MSSQL | POSTGRESQL
[--dbname db_name]
[--dbuser db_user]
[--dbpassword db_password]
[--dbport db_port]
[--dbhostname db_hostname]
[--dbdriverpath db_driver_path]
[--dbsslconnection true | false]

```

Db2 for z/OS-only configuration options

```
[--zlocationname zOS_location_containing_db]
```

SSL configuration options

```

--sslkeysfolder keystore_truststore_folder
--sslpassword ssl_password

```

User information

```
--user | -u dwc_user  
--password | -p dwc_password
```

Configuration information for the application server

```
--wlpdir|-w wlp_directory
```

Security configuration

```
[--httpsport https_port]  
[--bootstrapport bootstrap_port]  
[--bootsecpport bootstrap_sec_port]
```

Syntax for UNIX operating systems

Show command usage

```
dwcinst -? | --usage | --help
```

Retrieve the command parameters and values from a properties file

```
dwcinst --file | -f [properties_file]
```

General information

```
dwcinst  
--acceptlicense yes|no  
[--lang lang_id]  
[--inst_dir install_dir]  
[--data_dir dwc_datadir]  
[--skipcheckprereq true|false]  
[--componenttype DWC | FED]
```

Configuration information for the data source

```
--rdbmstype|-r DB2 | DB2Z | ORACLE | MSSQL | POSTGRESQL  
[--dbname db_name]  
[--dbuser db_user]  
[--dbpassword db_password]  
[--dbport db_port]  
[--dbhostname db_hostname]  
[--dbdriverpath db_driver_path]  
[--dbsslconnection true | false]
```

Db2 for z/OS-only configuration options

```
[--zlocationname zOS_location_containing_db]
```

SSL configuration options

```
--sslkeysfolder keystore_truststore_folder
--sslpassword ssl_password
```

User information

```
--user | -u dwc_user
--password | -p dwc_password
```

Configuration information for the application server

```
--wlpdir|-w wlp_directory
```

Security configuration

```
[--httpsport https_port]
[--bootstrapport bootstrap_port]
[--bootsecpport bootstrap_sec_port]
```

Syntax for z/OS operating systems**Show command usage**

```
dwcinst -? | --usage | --help
```

Retrieve the command parameters and values from a properties file

```
dwcinst --file | -f [properties_file]
```

General information

```
dwcinst
--acceptlicense yes|no
[--lang lang_id]
[--inst_dir install_dir]
[--data_dir dwc_datadir]
[--componenttype DWC | FED]
```

Configuration information for the data source

```
--rdbmstype|-r DB2 | DB2Z | ORACLE | MSSQL | POSTGRESQL
[--dbname db_name]
[--dbuser db_user]
[--dbpassword db_password]
[--dbport db_port]
[--dbhostname db_hostname]
```

```
[--dbdriverpath db_driver_path]
```

Db2 for z/OS-only configuration options

```
[--zlocationname zOS_location_containing_db]
```

SSL configuration options

```
--sslkeysfolder keystore_truststore_folder  
--sslpassword ssl_password
```

User information

```
--user | -u dwc_user  
--password | -p dwc_password
```

Configuration information for the application server

```
--wlpdir | -w wlp_directory
```

Security configuration

```
[--httpsport https_port]  
[--bootstrapport bootstrap_port]  
[--bootsecport bootstrap_sec_port]
```

Parameters

-? | -usage | -help

Displays the command usage and exits.

--propfile | -f [*properties_file*]

Optionally specify a properties file containing custom values for `dwcinst` parameters. The default file is located in the root directory of the installation image.

Specifying a properties file is suggested if you have a high number of parameters which require custom values. You can also reuse the file with minimal modification for several installations. If you create a custom properties file, specify its name and path with the **-f** parameter.

General information

--acceptlicense *yes/no*

Specify whether to accept the License Agreement.

--lang lang_id

The language in which the messages returned by the command are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used. If neither **--lang** nor LANG are used, the default codepage is set to SBCS. For a list of valid values for these variables, see the following table:

Table 27. Valid values for -lang and LANG**parameter**

Language	Value
Brazilian Portuguese	pt_BR
Chinese (traditional and simplified)	zh_CN, zh_TW
English	en
French	fr
German	de
Italian	it
Japanese	ja
Korean	ko
Russian	ru
Spanish	es



Note: This is the language in which the installation log is recorded and not the language of the installed component instance. The command installs all languages as default.

--inst_dir

Specify the directory where the Dynamic Workload Console is to be installed. This parameter is optional. The default values varies based on the operating system, as follows:

On Windows operating systems

```
%ProgramFiles%\wa\DWC
```

On UNIX operating systems

```
/opt/wa/DWC
```

On z/OS operating system

```
/opt/wa/DWC
```

After installing, you can find this value in the

twainstance<instance_number>.TWA.properties file, by checking the **DWC_basePath**

parameter. For more information, see the topic about Finding out what has been installed in which HCL Workload Automation instances in *Administration Guide*.

--data_dir *dwc_datadir*

Specify the path to a directory where you want to store the logs and configuration files produced by Dynamic Workload Console. This parameter is optional. If you do not specify this parameter, all data files generated by the Dynamic Workload Console are stored in *DWC_home/DWC_DATA*. This path is called, in the publications, *DWC_DATA_dir*.

--skipcheckprereq *true* | *false*

If you set this parameter to `true`, Dynamic Workload Console does not scan system prerequisites before starting the installation. This parameter is optional. The default value is `false`. For more information about the prerequisite check, see [Scanning system prerequisites for HCL Workload Automation on page 48](#).

Configuration information for the data source

--rdbmstype|-r *rdbms_type*

The database type. Supported databases are:

- **DB2**
- **ORACLE**
- **MSSQL** This value applies to MSSQL and supported MSSQL cloud-based databases.
- **POSTGRESQL**
-

This parameter is required and has no default value.

--dbname *db_name*

The name of the Dynamic Workload Console database. This parameter is optional. The default value is **DWC**.

--dbuser *db_user*

The user that has been granted access to the Dynamic Workload Console tables on the database server. This parameter is required.

--dbpassword *db_password*

The password for the user that has been granted access to the Dynamic Workload Console tables on the database server. This parameter is required. Special characters are not supported. You can optionally encrypt the password. For more information, see [Encrypting passwords \(optional\) on page 58](#).

--dbport *db_port*

The port of the database server. This parameter is required.

--dbhostname *db_hostname*

The host name or IP address of database server. This parameter is required.

--dbdriverpath *db_driver_path*

The path where the database drivers are stored. This parameter is optional, but becomes required when you set the **rdbmstype** parameter to **DB2Z**. If you use Db2 for z/OS with the Dynamic Workload Console version 10.2.4 or later, transfer the drivers in binary mode from the directory where you installed Db2 for z/OS to a directory of your choice. Specify the driver path using this parameter.

By default, the configuration script references the JDBC drivers supplied with the product images. If your database server is not compatible with the supplied drivers, then contact your database administrator for the correct version to use with your database server and specify the driver path using this parameter. Ensure you provide the same path in the `configureDb`, `serverinst`, and `dwcinst` commands.

--dbsslconnection true | false

Enables or disables the SSL connection to the database. This value must always be **false** when `--rdbmstype` is **DB2Z**.

The default value is **false**.

SSL configuration options**--sslkeyfolder *keystore_truststore_folder***

The name and path of the folder containing certificates in PEM format. The installation program automatically processes the keystore and truststore files using the password you specify with the `--sslpassword` parameter. The folder must contain the following files:

- **ca.crt**

The Certificate Authority (CA) public certificate. Note that if certificates being installed are part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then this file must contain the Root CA certificate only. Any Intermediate CA certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.



Note: From V10.2.3, if certificates being installed are part of a chain, the `ca.crt` can contain also the intermediate CAs. In this case, it must begin with one or more intermediate CA certificates and end with the Root ca.

- **tls.key**

The private key of the end user certificate for the instance to be installed.

- **tls.crt**

The public part of the previous key, that is the end user certificate.

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

You can optionally create a subfolder to contain one or more *.`cert` files to be added to the server truststore as trusted CA, whose name must be `additionalCAs`. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or DB2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**. Note that if the end user certificate being installed in the instance is part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then the Intermediate CAs certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

For further information about how to generate custom certificates, see the topic about managing certificates using Certman in *HCL Workload Automation: Planning and Installation*.

--sslpassword *ssl_password*

The password for the certificates.

For more information, see [sslkeysfolder on page 463](#).

You can optionally encrypt the password using the secure script. For more information, see [Optional password encryption - secure script on page 427](#).

--enablefips *false*

Specify whether you want to enable FIPS. The default value is `false`. This parameter is optional.

Db2 for z/OS-only configuration syntax

--zlocationname *zos_location_containing_db*

The name of an already existing location in the z/OS environment that will contain the new database. The default value is `LOC1`.

User information

--user

Specify the administrator of the Dynamic Workload Console. You can use this account to log in to the Dynamic Workload Console and manage your environment. This parameter is optional. The default value is `dwcadmin`.

--password

Specify the password for the Dynamic Workload Console user. This parameter is required. You can optionally encrypt the password. For more information, see [Encrypting passwords \(optional\) on page 58](#).

On Windows operating systems

Supported characters for the password are alphanumeric, dash (-), underscore (_) characters, and `()!*~+.@!^`

On UNIX operating systems

Supported characters for the password are any alphanumeric, dash (-), underscore (_) characters, and ()|?=*~+.

Configuration information for the application server**--wlpdir**

Specify the path where Open Liberty is installed. This parameter is required.

On z/OS operating system

Specify the path where WebSphere Application Server for z/OS Liberty is installed. This parameter is required.

Security configuration**--httpsport**

Specify the HTTPS port, to be used in the Dynamic Workload Console URL. This parameter is optional. The default value is 9443.

--bootstrapport

Specify the bootstrap port. This parameter is optional. The default value is 12809.

--bootsecport

Specify the bootstrap security port, to be used for connecting to the Z connector. This parameter is optional. The default value is 19402.

Agent installation parameters - twsinst script

Agent installation parameters that can be passed to the twsinst script.

About this task

This section lists and describes the parameters that are used when running a twsinst script to install , , or (also known as agent with z-centric capabilities).



Note: To install a on HCL Universal Orchestrator, see [Installing and connecting HCL Workload Automation dynamic agents at version 10.2.3 or later](#).

To find some sample agent installation scenarios, see [Example installation commands on page 132](#) and [Dynamic agent gateway installation examples on page 135](#).

To manage authentication and certificates effectively, see [Table 29: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 476](#) for a comprehensive list of supported combinations for the following parameters:

- [-apikey on page 467](#)
- [-jwt true | false on page 469](#)
- [-sslkeysfolder path on page 472](#)
- [-sslpassword password on page 473](#)
- [-tdwbhostname host_name on page 473](#)
- [-tdwbport tdwbport_number on page 473](#)
- [-wapassword wauser_password on page 474](#)
- [-wauser wauser_name on page 475](#)

-acceptlicense *yes/no*

Specifies whether to accept the License Agreement.

-addjruntime *true/false*

Adds the Java™ run time to run job types with advanced options, both those types that are supplied with the product and the additional types that are implemented through the custom plug-ins. Valid values are **true** and **false**. The default for a fresh installation is **true**. Set this parameter to `true` if you use the **sslkeysfolder** and **sslpassword** parameters to define custom certificates in PEM format.

If you decided not to install Java™ run time at installation time, you can still add this feature later as it is described in [Adding a feature on page 225](#).

-agent *dynamic/fta/both/zcentric*

Specifies the type of agent to install. Valid values are:

dynamic

Installs the dynamic agent. **Requires** the **-tdwbhostname** *host_name* and the **-tdwbport** *tdwbport_number* parameters.

fta

Installs the fault-tolerant agent.

both

Installs the dynamic agent that is used with the **-tdwbhostname** *host_name* and the **-tdwbport** *tdwbport_number* parameters, and a fault-tolerant agent.

zcentric

Installs the (also known as agent with z-centric capabilities).

The default is ***dynamic***.

-agentid *agent_id*

Specifies a unique identifier for the agent. If not provided, an alphanumeric ID is automatically generated

```
893164748CCA4FC6820F12685AECBB07
```

To reuse an agent ID for reinstallations, specify the same `agent_id`. Ensure the value is exactly 32 characters long; otherwise, an error occurs.

If you set the `jwt` parameter to `true`, the `agentId` parameter is ignored if provided, because the agent ID is retrieved from the together with the JWT. See [-jwt true | false on page 469](#).

-apikey

Specifies the API key for authentication with the . This key enables downloading certificates or JWT for communication between and . A random password in base64 encoding is automatically created for generating stash files. The password stored in the `tls.sth` file. If needed, you can decrypt this password using any base64 decoder.

Obtain the string to be provided with this parameter from the before running the command. For more information, see the section about authenticating the command line client using API Keys in .

This parameter is **mutually exclusive** with:

- [-wauser wauser_name on page 475](#)
- [-wapassword wauser_password on page 474](#)
- [-sslkeyfolder path on page 472](#)
- [-sslpassword password on page 473](#)

and it is **required** with:

- [-tdwbhostname host_name on page 473](#)
- [-tdwbport tdwbport_number on page 473](#)

For a comprehensive list of supported combinations for this parameter and related ones, see [Table 29: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 476](#).

-company company_name

The name of the company. The company name cannot contain blank characters. The name is shown in program headers and reports. If not specified, the default name is COMPANY.

-create_link

UNIX™ systems only. Create the **symlink** between `/usr/bin/at` and `install_dir/TWS/bin/at`. For more information, see [Table 2: Symbolic link options on page 34](#).

-data_dir path

This argument applies to UNIX operating systems only. Specify a path for product data, such as log and configuration files, if you want to install the product binaries separated from the product data. This argument is optional. The default value is `INSTALL_DIR/TWSDATA`.

-displayname *display_name*

The name to assign to the agent. The name cannot start with a number. The default is based on the host name of this computer.

If the host name starts with a number, the **-displayname** parameter must be specified.

-domain *user_domain*

Windows™ systems only. The domain name of the user. The default is the name of the workstation on which you are installing the product. Ensure you use `USERDOMAIN` instead of `USERDNSDOMAIN`.

-enablefips *true/false*

Specify whether you want to enable FIPS. The default value is `false`. This parameter is optional.

-encryptionpassword *default***-gateway *local/remote/none***

Specifies whether to configure a gateway to communicate with the or not, and how it is configured. Specify `local` if the gateway is local to the workstation. Specify `remote` if the communicates through a gateway that is installed on a different workstation from the being installed. The default value is `none`, which means no gateway is configured. For information about installing with a local and remote gateway, see [Example installation commands on page 132](#).

-gweifport *gateway_eif_port*

Specifies the Job Manager Event Integration Facility (EIF) port number. The default value is **31132**. The valid range is 1 to 65535.

-gwid *gateway_id*

The unique identifier for the gateway. This parameter is required when you specify **-gateway *local*** and must be unique across all agents. The default gateway identifier that is assigned is **GW1**. The gateway identifier must start with either an alphabetic character or an underscore character (`_`), and it can contain only the following types of characters: alphabetic, numeric, underscores (`_`), hyphens (`-`), and periods (`.`).

Gateways can also work in parallel to mutually take over in routing communications to the agents connected to them. To enable gateways to work in parallel, all gateways must have the same *gateway_id* assigned. This information is stored in the `JobManagerGW.ini` file, by setting the **JobManagerGWURIs** property.

-hostname *host_name*

The fully qualified hostname or IP address on which the agent is contacted by the . The default is the hostname of this computer. If the hostname is a localhost, the hostname parameter must be specified.

-inst_dir *installation_dir*

The directory of the installation.

On Windows™ operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. Specify an absolute path. If you do not manually specify a path, the path is set to `%ProgramFiles%\HCL`

`\TWA_TWS_USER`, where `TWS_USER` is the user for which you are installing the that you specify in the `-uname` parameter. If you use the Local System Account and therefore do not specify the `-uname` parameter, the path is set to `%ProgramFiles%\HCL\TWA_WaLocalSystemAccount`.

On UNIX™ and Linux™ operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. Specify an absolute path. If you do not manually specify a path, the path is set to:

- `/opt/HCL/TWA_TWS_USER`, if you logged in as the **root** user to install the agent. `TWS_USER` is the user that you specify in the `-uname` option and for which you are installing the agent (can omit if `TWS_USER` is **root**). The user that you specify in the `-uname username` parameter must have read and run privileges for the `installation_dir` installation path; otherwise the installation fails.
- `home_dir/TWA`, if you logged in with a login **other than root**. Ensure that the directory permission is set to **755** for `home_dir`, the home directory for your login, and that you are the `home_dir` owner.

-jimport port_number

The JobManager port number used by the to connect to the . The default value is **31114**. The valid range is from 1 to 65535.

-jimportssl true/false

The JobManager port used by the to connect to the . The port value is the value of the `ssl_port` parameter in the `ita.ini` file if **-jimportssl** is set to `true`. If set to `false`, it corresponds to the value of the `tcp_port` parameter in the `ita.ini` file. The `ita.ini` file is located in `ITA\cpa\ita` on Windows™ systems and `ITA/cpa/ita` on UNIX™, Linux™, and systems.

Set the value to "true" if **- gateway** is set to `local`.

For communication using SSL or HTTPS

Set **jimportssl = true**. To communicate with the , it is recommended that you set the value to `true`. In this case, the port specified in **jimport** communicates in HTTPS.

For communication without using SSL or through HTTP

Set **jimportssl = false**. In this case the port specified in **jimport** communicates in HTTP.

-jwt true / false

Specify `true` to use a JSON Web Token (JWT) for authentication with the . Specify `false` to authenticate with the using certificates instead. The default value is `true`.

When set to `true`, this parameter is **mutually exclusive** with the following parameters which are used to generate custom certificates:

- [-sslkeysfolder path on page 472](#)
- [-sslpassword password on page 473](#)

If you set this parameter to `true`, note the following:

- the `-agentid agent_id` on [page 466](#), if provided, will be ignored because the agent ID is retrieved from the along with the JWT.
- The following parameters are **required** for downloading the JWT:
 - `-wauser wauser_name` on [page 475](#) or `-apikey` on [page 467](#).
 - `-wapassword wauser_password` on [page 474](#) or `-apikey` on [page 467](#).
 - `-tdwbhostname host_name` on [page 473](#). This parameter is always required when `jwt` is set to `true`, regardless of whether you use the **wauser** and **wapassword** or the **apikey** parameters.
 - `-tdwbport tdwbport_number` on [page 473](#). This parameter is always required when `jwt` is set to `true`, regardless of whether you use the **wauser** and **wapassword** or the **apikey** parameters.

For examples of installations with JWT, see [Example installation commands on page 132](#).

For a comprehensive list of supported combinations for this parameter and related ones, see [Table 29: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 476](#).

-lang lang_id

The language in which the twsinst messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used. If neither **-lang** nor LANG are used, the default codepage is set to SBCS. For a list of valid values for these variables, see the following table:

Table 28. Valid values for -lang and LANG
parameter

Language	Value
Brazilian portuguese	pt_BR
Chinese (traditional and simplified)	zh_CN, zh_TW
English	en
French	fr
German	de
Italian	it
Japanese	ja
Korean	ko
Russian	ru

Table 28. Valid values for -lang and LANG**parameter (continued)**

Language	Value
Spanish	es



Note: This is the language in which the installation log is recorded and not the language of the installed engine instance. twsinst installs all languages as default.

-master workstation

The workstation name of the master domain manager. This name cannot exceed 16 characters, cannot contain spaces, and cannot be the same as the workstation name that you entered in the **thiscpu** parameter. If not specified, the default value is **MASTER**.

-netmansslport SSL_port_number**-new**

A fresh installation of the agent. Installs an agent and all supported language packs.

-password user_password

Windows™ systems only. The password of the user for which you are installing . The password can include alphanumeric, dash (-), and underscore (_) characters, and the following symbols: (!)?=^*/~ [] \$`+;.:@. The **-password** parameter is used for fresh installations only, it is not required for fix packs or upgrades. You can optionally encrypt the password using the secure script. For more information, see [Optional password encryption - secure script on page 427](#).

-port port_number

The TCP/IP port number used by the Netman process to listen for communication from the master. The default value is **31111**. The valid range is from 1 to 65535. This port number is registered in the `localopts` file. For each installation you must specify a different number.

-reset_perm

UNIX™ and systems only. Reset the permission of the libraries in the `/usr/hcl` directory.

-restore

Run this command from the folder to where you copied the elmage (a folder other than the home directory of `TWS_USER`, where `TWS_USER` is the user that installed the instance), and not from the installation path, to restore the version in the elmage.

-skip_usercheck

Enable this option if the authentication process within your organization is not standard, thereby disabling the default authentication option.

On Windows™ systems, if you specify this parameter, the program does not create the user you specified in the **-uname** *username* parameter and you must create the user manually before running the script. However, if you use Local System Account, you do not need to specify any user.

On UNIX™ and Linux™ systems if you specify this parameter, the program skips the check of the user in the `/etc/passwd` file or the check you perform using the `su` command.

-skipcheckprereq

If you specify this parameter, does not scan system prerequisites before installing the agent. For more information on the prerequisite check, see [Scanning system prerequisites for HCL Workload Automation on page 48](#).

-sslkeyfolder *path*

The name and path of the folder on the agent containing PEM certificates. The installation program automatically generates the keystore and truststore files using the password you specify with the **sslpassword** parameter, which is **required** when using **sslkeyfolder**.

The folder must contain the following files and folders:

ca.crt

The Certificate Authority (CA) public certificate.

tls.key

The private key for the instance to be installed.

tls.crt

The public part of the previous key.

tls.sth

The file storing your encoded password in Base64 encoding.

You can optionally create a subfolder to contain one or more `*.crt` files to be added to the server truststore as trusted CA. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or DB2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**. If you are connecting a using custom certificates to a also using custom certificates, the only required file is `ca.crt`.

Before you start the installation, ensure the required files and folders are available on the agent.

The **sslkeyfolder** and **sslpassword** parameters are **mutually exclusive** with the **wauser**, **wapassword**, **apikey**, and **jwt true** parameters, which are used to download and deploy the certificates or JWT already available on the .

For a comprehensive list of supported combinations for this parameter and related ones, see [Table 29: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 476](#).

-sslpassword *password*

Specify the password for the certificates in PEM format automatically generated by the installation program. It requires the **sslkeysfolder** parameter.

If you use this parameter, ensure that the **addjruntime** parameter is set to true, because Java™ run time is required for defining custom certificates.

The **sslkeysfolder** and **sslpassword** parameters are **mutually exclusive** with the **wauser**, **wapassword**, **apikey**, and **jwt true** parameters, which are used to download and deploy the certificates or JWT already available on the .

For a comprehensive list of supported combinations for this parameter and related ones, see [Table 29: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 476](#)

-tdwbhostname *host_name*

The fully qualified host name or IP address of the the agent is registering to. This parameter **requires** the **-tdwbport** parameter. It **is required** if you use the **wauser** and **wapassword** or the **apikey** parameters. This parameter is not supported on (also known as agent with z-centric capabilities).

If you set the **-gateway** parameter to `remote`, this is the host name of the hosting the gateway and to which the agent you are installing will connect. This information is stored in the `JobManager.ini` file. For information about installing with a local and remote gateway, see [Example installation commands on page 132](#).

See also [-jwt true | false on page 469](#).

For a comprehensive list of supported combinations for this parameter and related ones, see [Table 29: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 476](#).

-tdwbport *tdwbport_number*

The HTTPS transport port number of the the agent is registering to. It must match the port you specified with **httpsport** parameter when installing the . It is **required** if you use the **wauser** and **wapassword** or the **apikey** parameters and **requires** the **-tdwbhostname** parameter. This parameter is not supported on (also known as agent with z-centric capabilities).

The valid range is from 0 to 65535. If you specify 0 you cannot run workload dynamically. Do not specify 0 if the **-agent** value is `dynamic` or `both`. The default is 0 for an upgrade, which means that this connection is not configured, otherwise, specify 31116 for a fresh installation.

If you set the **-gateway** parameter to `remote`, this is the HTTP or HTTPS port number of the hosting the gateway and to which the agent you are installing will connect. You have specified this port with the **jimport** parameter when installing the agent hosting the gateway. For information about installing with a local and remote gateway, see [Example installation commands on page 132](#).

If you are performing a fresh installation, the value to use is `31114`. This information is stored in the `JobManager.ini` file.

For a comprehensive list of supported combinations for this parameter and related ones, see [Table 29: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 476](#).

-thiscpu workstation

The name of the workstation of this installation. The name cannot exceed 16 characters, cannot start with a number, cannot contain spaces, and cannot be the same as the workstation name of the master domain manager. This name is registered in the `localopts` file. If not specified, the default value is the host name of the workstation.

If the host name starts with a number, **-thiscpu** parameter must be specified.

-u

Displays command usage information and exits.

-uname username

The name of the user for which the agent is being installed. This user owns the instance and by default, jobs are run with its name. This user name is not to be confused with the user performing the installation, unless you use a **user other than root**. The user name cannot contain periods (.).

On UNIX™ and Linux™ systems, for a new installation, this user account must be created manually before running the installation and must be enabled to login to the machine where the agent is going to be installed. Create a user with a home directory. is installed by default under the home directory of the specified user.

On Windows operating systems, you can install and using the Local System Account by omitting the **uname** and **password** parameters.

-useencryption true | false

-v

Displays the command version and exits.

-wapassword wouser_password

One of the following passwords, defined on the :

- The password of the user for which you have installed the the agent is connecting to.
- The password of the user with the DISPLAY permission on the FILE named AGENT_CERTIFICATE. This permission allows the user to download certificates or JWT. For more information about this scenario, see [Downloading certificates or JWT using a different user on page 487](#).

See also [-jwt true | false on page 469](#).

For further information about how to automatically download and deploy certificates in PEM format from the master domain manager to dynamic agents and fault-tolerant agents, see [Certificates download to dynamic agents and fault-tolerant agents - AgentCertificateDownloader script on page 482](#).

This parameter always requires the [tdwbport on page 473](#) and [-tdwbhostname host_name on page 473](#) parameters.

For a comprehensive list of supported combinations for this parameter and related ones, see [Table 29: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 476](#).

-wuser wuser_name

One of the following users, defined on the :

- The user for which you have installed the the agent is connecting to.
- The user with the DISPLAY permission on the FILE named AGENT_CERTIFICATE. This permission allows the user to download certificates or JWT. For more information about this scenario, see [Downloading certificates or JWT using a different user on page 487](#).

Always specify the user defined on the , also if you are installing a and want it to register to a . This is because the simply forwards data to and from the .

By providing the **wuser** and **wpassword** parameters or the **apikey** parameter, you enable to download and install either the certificates or the JWT already available on the :

- To download certificates, set the **jwt** parameter to `false`
- To download JWT, set the **jwt** parameter to `true`. For more information, see [-jwt true | false on page 469](#).

Key details about this parameter:

- It is **mutually exclusive** with the [-apikey on page 467](#) parameter, which provides authentication using an API Key and the [-sslkeyfolder path on page 472](#) and [-sslpassword password on page 473](#) parameters.
- It **always requires** the [tdwbport on page 473](#) and [-tdwbhostname host_name on page 473](#) parameters.
- It is **not supported** on the (also known as the agent with z-centric capabilities). To generate certificates for the , use the **sslkeyfolder** and **sslpassword** parameters.

For further information about how to automatically download and deploy certificates in PEM format from the master domain manager to dynamic agents and fault-tolerant agents, see [Certificates download to dynamic agents and fault-tolerant agents - AgentCertificateDownloader script on page 482](#).

For a comprehensive list of supported combinations for this parameter and related ones, see [Table 29: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 476](#).

-work_dir *working_dir*

The temporary directory used by the program to deploy the installation process files.

On Windows™ operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. If you do not manually specify a path, the path is set to %temp%\TWA*twsversion_number*, where %temp% is the temporary directory of the operating system.

On UNIX™ and Linux™ operating systems:

The path cannot contain blanks. If you do not manually specify a path, the path is set to /tmp/*TWA/twsversion_number*.

This parameter can also function as a backup directory during product upgrade with path *WORKING_DIR/backup* if you do not set the **-skipbackup** parameter to **true**.

Table 29. Supported combinations and mutual exclusions for authentication and certificate-related parameters

Parameter	Compatibility	Mutual exclusion	Required with
-apikey	Used to authenticate with the master domain manager and download certificates or JWT.	Cannot be used with: <ul style="list-style-type: none">• -wauser• -wapassword• -sslkeysfolder• -sslpassword	<ul style="list-style-type: none">• -tdwbhostname• -tdwbport
-jwt false	Uses certificates for authentication.	No direct exclusion	<div>EITHER<ul style="list-style-type: none">• -apikey• -tdwbhostname• -tdwbportOR<ul style="list-style-type: none">• -wauser• -wapassword• -tdwbhostname• -tdwbport</div>

Table 29. Supported combinations and mutual exclusions for authentication and certificate-related parameters (continued)

Parameter	Compatibility	Mutual exclusion	Required with
			OR • -sslkeysfolder • -sslpassword
-jwt true	Uses JWT for authentication.	Cannot be used with: • -sslkeysfolder • -sslpassword	EITHER • -apikey • -tdwbhostname • -tdwbport OR • -wuser • -wapassword • -tdwbhostname • -tdwbport
-sslkeysfolder	Specifies a folder containing PEM certificates.	Cannot be used with • -apikey • -wuser • -wapassword • -jwt true	• -sslpassword
-sslpassword	Password for PEM certificates.	Cannot be used with • -apikey • -wuser • -wapassword • -jwt true	• -sslkeysfolder
-tdwbhostname	Hostname/IP of the Dynamic Workload Broker (DWB).	No direct exclusion	EITHER • -apikey • -tdwbport OR

Table 29. Supported combinations and mutual exclusions for authentication and certificate-related parameters (continued)

Parameter	Compatibility	Mutual exclusion	Required with
			<ul style="list-style-type: none"> • -wauser • -wapassword • -tdwbport
-tdwbport	HTTPS port of the Dynamic Workload Broker (DWB).	No direct exclusion	EITHER <ul style="list-style-type: none"> • -tdwbhostname • -apikey OR <ul style="list-style-type: none"> • -tdwbhostname • -wauser • -wapassword
-wauser	User for authentication and downloading certificates/JWT	Cannot be used with <ul style="list-style-type: none"> • -apikey • -sslkeysfolder • -sslpassword 	<ul style="list-style-type: none"> • -wapassword • -tdwbhostname • -tdwbport
-wapassw ord	.Password of the -wauser user.	Cannot be used with <ul style="list-style-type: none"> • -apikey • -sslkeysfolder • -sslpassword 	<ul style="list-style-type: none"> • -wauser • -tdwbhostname • -tdwbport

File Proxy installation - fileproxyinst script

This script installs the File Proxy in SSL mode

This section lists and describes the parameters that are used when running the **fileproxyinst** script to install the File Proxy in SSL mode on a workstation different from the master domain manager, where it is already installed by default. This command is supported on the following operating systems:

- Windows with hardware x86-64
- Linux with hardware x86-64
- Linux with hardware IBM z Systems

You can optionally install the File Proxy in high availability configuration by specifying one or more proxy servers or a load balancer. To set up this configuration, use the **Broker.fileproxy.urls** property in the `BrokerWorkstation.properties`. For more information, see the topic about the `BrokerWorkstation.properties` file in *User's Guide and Reference*.

Log files produced by this command are located in `data_dir/logs/`. By default `data_dir` is

`installation_directory/FILEPROXYDATA`.

Syntax

Windows operating systems:

```
fileproxyinst.exe -acceptlicense yes [-lang language] -inst_dir installation_directory
[-data_dir data_directory]
[-host hostname] [-sslport ssl_port_number] -sslfolder ssl_folder -sslpassword ssl_pwd
[-java_home java_home]
```

Linux64 and Linux for OS/390 operating systems:

```
./fileproxyinst -acceptlicense yes [-lang language] -inst_dir installation_directory
[-data_dir data_directory]
[-host hostname] [-sslport ssl_port_number] -sslfolder ssl_folder -sslpassword ssl_pwd
[-java_home java_home]
```

Arguments

-acceptlicense *yes|no*

Required. Specify whether to accept the License Agreement. The default is `no`.

-lang *language*

Optional. The language in which the messages returned by the command are displayed. The default value is `en_us`.

-inst_dir *installation_directory*

Required. The directory of the File Proxy installation. The default value is the directory from which you start the command. Ensure that you have write access to this directory.

-data_dir *data_dir*

Optional. The directory where logs and configuration files are stored. The default value is `installation_directory/FILEPROXYDATA`. Ensure that you have write access to this directory.

-host *hostname*

Optional. The global, public host name of the workstation where you install the File Proxy or the IP address of the workstation.

-sslport *ssl_port_number*

Optional. The port to be used for secure communication. Supported values are integers between 1 and 65535. The default is `44444`.

-sslfolder *ssl_folder*

Required. The name and path of the folder containing the certificates in PEM format. (For details about how to create the certificates, see the section about creating a certificate authority in the *HCL Workload Automation: Administration Guide*). It must contain the following files and folder:

- `ca.crt`
- `tls.crt`
- `tls.key`
- `additionalCAs` folder



Note: In the z/OS environment:

- When you create the certificate authority by issuing the command `./openssl x509 -req -in tls.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out tls.crt -days xxx`, ensure that you set the appropriate number of days with the `-days` parameter. The default is 30.
- The `additionalCAs` folder must contain the public key certificate or public certificate chain of the Z controller SSL keyring.

-sslpassword *ssl_pwd*

Required. The password to access the certificates.

-java_home *java_home*

Optional. The default value is the path to the `.jre` file provided with the product image.

Examples

To specify the installation and data directories and also define a value for the port to be used for SSL communication, for the folder where the certificates are stored, and the password to access them, run the following command:

```
./fileproxyinst -acceptlicense yes -inst_dir /opt/wa/fileproxy -data_dir /opt/wa/mydata -sslport 44445
-sslfolder /opt/wa/my_ssl_folder -sslpassword my_ssl_pwd
```

To specify only the required parameters for the command, defining the folder where the certificates are stored, and the password to access them, run the following command:

```
./fileproxyinst -acceptlicense yes -sslfolder /opt/wa/my_ssl_folder -sslpassword my_ssl_pwd
```

File Proxy start - fileproxystart script

This script starts the File Proxy in SSL mode

This section describes the **fileproxystart** script. This script is used to start the File Proxy in SSL mode on a workstation different from the master domain manager, where it is already installed by default. Use this command to start the File Proxy in case it stops unexpectedly. This command is supported on the following operating systems:

- Windows with hardware x86-64
- Linux with hardware x86-64
- Linux with hardware IBM z Systems

By default, log files produced by this command are located in *data_dir/logs*.

Syntax

Windows operating systems:

```
fileproxystart.exe
```

Linux:

```
./fileproxystart
```

File Proxy stop - fileproxystop script

This script stops the File Proxy

This section describes the **fileproxystop** script. This script is used to stop the File Proxy. This command is supported on the following operating systems:

- Windows with hardware x86-64
- Linux with hardware x86-64
- Linux with hardware IBM z Systems

Syntax

Windows operating systems:

```
fileproxystop.exe
```

Linux:

```
./fileproxystop
```

File Proxy uninstallation - uninstall script

This script uninstalls the File Proxy

This section describes the **uninstall** script. This script is used to uninstall the File Proxy. This command is supported on the following operating systems:

- Windows with hardware x86-64
- Linux with hardware x86-64
- Linux with hardware IBM z Systems

By default, log files produced by this command are located in *data_dir/logs*.

Launch the command from the installation directory.

Syntax

Windows operating systems:

```
uninstall.exe -inst_dir installation_directory [-lang language]
```

Linux:

```
./uninstall -inst_dir installation_directory [-lang language]
```

where

-inst_dir *installation_directory*

is the directory where the File Proxy is installed.

-lang *language*

is the language in which the messages returned by the command are displayed.

Comments

The command removes all data related to the File Proxy and leaves the *data_dir* unchanged.

Certificates download to dynamic agents and fault-tolerant agents - AgentCertificateDownloader script

This script downloads and deploys certificates in PEM format from the master domain manager to dynamic agents and fault-tolerant agents or enables authentication through JSON Web Token (JWT).

You can use this script either to download and deploy certificates in PEM format from the master domain manager to the dynamic agents and fault-tolerant agents in your environment or to enable authentication through JSON Web Token (JWT).

This script does not apply to the HCL Workload Automation Agent (also known as agent with z-centric capabilities)

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* or the **apikey** parameters. The certificates in PEM format are automatically downloaded and deployed to the agent without further intervention. The same happens if you are using JWT as authentication method.



Note: If you use a load balancer between the dynamic agent and the master domain manager, and the load balancer uses a Certificate Authority (CA), you must convert the .p12 certificates into the PEM format and replace the existing ones.

When certificates are nearing their expiration time, new certificates are automatically downloaded to agents, but error conditions might happen, for example in case the agent is down or disconnected when the automatic certificate update takes place. In this case, you can use this command if you need to change the PEM certificates after their expiration time, or if you have downloaded wrong PEM certificates.

You can also use this command to obtain a new JWT for an agent from which you had previously revoked the JWT. For more information about revoking a JWT, see [Revoking and reissuing a JSON Web Token on page 219](#).

If you authenticate using PEM certificates, the script connects to the master domain manager to retrieve the compressed file containing the certificates, and saves them to the working directory with name `waCertificates.zip`.

Before running the command, ensure the certificates in PEM format are available on the master domain manager in one of the following paths:

On Windows operating systems

```
installation_directory\TWS\ssl\depot
```

On UNIX operating systems

```
TWA_DATA_DIR/ssl/depot
```

The required files are:

ca.crt

The Certificate Authority (CA) public certificate.

tls.key

The private key for the instance to be installed.

tls.crt

The public part of the previous key.

tls.sth

The file storing your encoded password in Base64 encoding.

You can optionally create a subfolder to contain one or more `*.crt` files to be added to the server truststore as trusted CA. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or DB2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**. If you are connecting a master domain manager using custom certificates to a dynamic agent also using custom certificates, the only required file is `ca.crt`.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

After running the command, stop and restart the agent process using the `ShutDownLwa` and `StartUpLwa` commands. For more information about the commands, see `ShutDownLwa` - Stop the agent and `StartUpLwa` - Start the agent. For more information about the commands, see the sections about the `ShutDownLwa` and `StartUpLwa` commands in *User's Guide and Reference*.

Ensure you start the command from a brand-new shell.

Syntax

Certificate installation syntax on Windows operating systems

Show command usage

```
cscript AgentCertificateDownloader.vbs -? | --usage | --help
```

Retrieve the command parameters and values from a properties file

```
cscript AgentCertificateDownloader.vbs --file | -f [properties_file]
```

General information

```
cscript AgentCertificateDownloader.vbs
--wauser wauser_name
--wapassword wauser_password
[--jwt true]
[--apikey API_key_string]
[--tdwbhostname host_name]
[--tdwbport tdwbport_number]
--work_dir working_dir
[--gateway local | remote | none]
[--gwid gateway_id]
[--displayname agent_name]
```

Certificate installation syntax on UNIX operating systems**Show command usage**

```
./AgentCertificateDownloader.sh --? | --usage | --help
```

Retrieve the command parameters and values from a properties file

```
./AgentCertificateDownloader.sh --file | --f [properties_file]
```

General information

```
./AgentCertificateDownloader.sh
--wauser wauser_name
--wapassword wauser_password
[--jwt true]
[--apikey API_key_string]
[--tdwbhostname host_name]
[--tdwbport tdwbport_number]
--work_dir working_dir
[--gateway local | remote | none]
[--gwid gateway_id]
[--displayname agent_name]
```

AgentCertificateDownloader parameters

--? | --usage | --help

Displays the command usage and exits.

--propfile | --f [*properties_file*]

Optionally specify a properties file containing custom values for AgentCertificateDownloader parameters. The default file is located in the root directory of the installation image.

Specifying a properties file is suggested if you have a high number of parameters which require custom values. You can also reuse the file with minimal modification for several installations. If you create a custom properties file, specify its name and path with the **-f** parameter.

General information

--wauser *wauser_name*

One of the following users, defined on the master domain manager:

- The user for which you have installed the master domain manager the agent is connecting to.
- The user with the DISPLAY permission on the FILE named AGENT_CERTIFICATE. This permission allows the user to download certificates or JWT. For more information about this scenario, see [Downloading certificates or JWT using a different user on page 487](#).

By providing the **wauser** and **wapassword** parameters or the **apikey** parameter, you enable HCL Workload Automation to download and install either the certificates or the JWT already available on the master domain manager:

This parameter is always required, unless you specify the **apikey** parameter, which defines a different authentication method.

--wapassword *wauser_password*

One of the following passwords, defined on the master domain manager:

- The password of the user for which you have installed the master domain manager the agent is connecting to.
- The password of the user with the DISPLAY permission on the FILE named AGENT_CERTIFICATE. This permission allows the user to download certificates or JWT. For more information about this scenario, see [Downloading certificates or JWT using a different user on page 487](#).

Always specify the user defined on the master domain manager, also if you are installing a dynamic agent and want it to register to a dynamic domain manager. This is because the dynamic domain manager simply forwards data to and from the master domain manager.

By providing the **wauser** and **wapassword** parameters or the **apikey** parameter, you enable HCL Workload Automation to download and install either the certificates or the JWT already available on the master domain manager:

This parameter is always required, unless you specify the **apikey** parameter, which defines a different authentication method.

--jwt *true*

Specify `true` to download the JSON Web Token (JWT) to authenticate with the master domain manager. If you specified `true` at installation time, you can no longer change this setting to `false` and switch to PEM certificates. The only supported operation is switching from PEM certificates to JWT authentication by setting this parameter to `true`.

This parameter is mutually exclusive with the **sslkeysfolder** and **sslpassword** parameters which are used to generate custom certificates.

If you set this parameter to `true`, the following parameters are required for downloading the JWT:

- **wauser** or **apikey**
- **wapassword** or **apikey**
- **tdwbhostname**
- **tdwbport**

-apikey

Specifies the API key for authentication with the master domain manager. This key enables downloading certificates or JWT for communication between dynamic agent and dynamic domain manager. A random password in base64 encoding is automatically created for generating stash files. The password stored in the `tls.sth` file. If needed, you can decrypt this password using any base64 decoder.

Obtain the string to be provided with this parameter from the Dynamic Workload Console before running the command. For more information, see the section about authenticating the command line client using API Keys in *Dynamic Workload Console User's Guide*.

This parameter is **mutually exclusive** with:

- [-wauser wauser_name on page 129](#)
- [-wapassword wauser_password on page 129](#)
- [-sslkeysfolder path on page 126](#)
- [-sslpassword password on page 127](#)

and it is **required** with:

- [-tdwbhostname host_name on page 127](#)
- [-tdwbport tdwbport_number on page 127](#)

For a comprehensive list of supported combinations for this parameter and related ones, see [Table 7: Supported combinations and mutual exclusions for authentication and certificate-related parameters on page 130](#).

--tdwbhostname *host_name*

The fully qualified host name or IP address of the broker server to which the agent is connected. The default value is `localhost`. If you set the **jwt** parameter to `true`, ensure you provide a value for this parameter, to download the JWT and agent ID from the dynamic domain manager. The dynamic domain manager routes the JWT request to the master domain manager.

--tdwbport *tdwbport_number*

Specify the port of the broker server to which the agent is connected. The default value is 31116. If you set the **jwt** parameter to `true`, ensure you provide a value for this parameter, to download the JWT and agent ID from the dynamic domain manager. The dynamic domain manager routes the JWT request to the master domain manager.

--work_dir *working_dir*

The working directory used to store the `waCertificates.zip` file returned by the command. This compressed file contains the certificates in PEM format retrieved from the master domain manager. This parameter is required and no default value is provided.

--gateway *local/remote/none*

Specifies whether to configure a gateway to communicate with the dynamic workload broker or not, and how it is configured. Specify *local* if the gateway is local to the dynamic agent workstation. Specify *remote* if the dynamic agent communicates through a gateway that is installed on a different dynamic agent workstation from the dynamic agent being installed. The default value is *none*, which means no gateway is configured. For information about installing with a local and remote gateway, see [Example installation commands on page 132](#).

--gwid *gateway_id*

The unique identifier for the gateway. This parameter is required when you specify **-gateway *local*** and must be unique across all agents. The default gateway identifier that is assigned is **GW1**. The gateway identifier must start with either an alphabetic character or an underscore character (`_`), and it can contain only the following types of characters: alphabetic, numeric, underscores (`_`), hyphens (`-`), and periods (`.`).

Gateways can also work in parallel to mutually take over in routing communications to the agents connected to them. To enable gateways to work in parallel, all gateways must have the same *gateway_id* assigned. This information is stored in the `JobManagerGW.ini` file, by setting the **JobManagerGWURLs** property.

--displayname *agent_name*

Specify the name assigned the agent.

You can also use the **wapassword** and **wauser** parameters to specify a user different from the user which installed the master domain manager by using an ACL, as described in [Downloading certificates or JWT using a different user on page 487](#).

For more information about the typical installation procedure, see [Typical installation scenario on page 54](#).

Downloading certificates or JWT using a different user

Procedure to download and deploy certificates or JWT from the master domain manager to agents using a user different from the user which installed the master domain manager.

About this task

To define a user different from the user which installed the master domain manager, perform the following steps:

1. Browse to the `authentication_config.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`

2. Create a backup copy of the file to a different directory and add the new user and password to the file in the `overrides` directory.
3. Create a new role for the user, as follows:

```
composer new srol
SECURITYROLE DOWNLOAD_CERT_SROLE
FILE DISPLAY
END
```

4. Create a new domain for the user, as follows:

```
composer new sdom
SECURITYDOMAIN DOWNLOAD_DOMAIN
FILE NAME="AGENT_CERTIFICATE"
END
```

5. Create a new access control list for the user, as follows:

```
composer new acl
ACCESSCONTROLLIST FOR DOWNLOAD_DOMAIN
other_user DOWNLOAD_CERT_SROLE
END
```

where *other_user* is the user inserted into `authentication_config.xml`.

Result

You can now use the *other_user*, which has only the DISPLAY role for file `AGENT_CERTIFICATE`, to install the agent and download certificates or JWT, or to run the `AgentCertificateDownload` script and download and deploy certificates or JWT.

You can also perform the same operations from the Dynamic Workload Console, as described in the section about managing workload security in *Dynamic Workload Console User's Guide*.

Notices

This document provides information about copyright, trademarks, terms and conditions for product documentation.

© Copyright IBM Corporation 1993, 2016 / © Copyright HCL Technologies Limited 2016, 2025

This information was developed for products and services offered in the US. This material might be available from HCL 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.

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

HCL 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:

HCL

330 Potrero Ave.

Sunnyvale, CA 94085

USA

Attention: Office of the General Counsel

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

HCL

330 Potrero Ave.

Sunnyvale, CA 94085

USA

Attention: Office of the General Counsel

HCL TECHNOLOGIES LTD. 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. HCL 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-HCL 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 HCL product and use of those websites is at your own risk.

HCL 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:

HCL

330 Potrero Ave.

Sunnyvale, CA 94085

USA

Attention: Office of the General Counsel

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 HCL under terms of the HCL Customer Agreement, HCL 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-HCL products was obtained from the suppliers of those products, their published announcements or other publicly available sources. HCL has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-HCL products. Questions on the capabilities of non-HCL 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 HCL, 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. HCL, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. HCL shall not be liable for any damages arising out of your use of the sample programs.

© (HCL Technologies Limited) (2025).

Portions of this code are derived from Sample Programs.

© Copyright 2016

Trademarks

HCL®, and other HCL graphics, logos, and service names including "hcltech.com" are trademarks of HCL. Except as specifically permitted herein, these Trademarks may not be used without the prior written permission from HCL. All other trademarks not owned by HCL that appear on this website are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by HCL.

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 HCL 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 HCL.

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 HCL.

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.

HCL 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 HCL, 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.

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

Index

Numerics

- 9.5 environment
- with custom certificates 239, 258

A

- accessibility ix
- accessing
 - cluster
 - environment 179
- add
 - option to add the dynamic workload broker
 - resource command with twsinst 225
- adding
 - new features 225
- AES encryption algorithm 427
- agent 115, 176
 - dynamic agent 176
 - for distributed environment 115, 176
 - for end-to-end environment 115
 - how to uninstall manually 410
 - installation 113
 - installing for different user 487
- Agent 182
- agent certificates
 - managing 482
- agent dynamic 115, 176, 176
 - on 115
- agent fault-tolerant
 - static environment 13
- agent installation 119, 465
 - scanning system prerequisites 48, 236
 - user other than root user 115
- agent installation method
 - serverinst 43
- agent installation return code
 - twsinst 400
- agent is in running status
 - Centralized agent update 310, 310
 - update does not complete 310
- agent restore return code
 - twsinst 400
- agent security
 - PEM certificates 482
- agent uninstallation return code
 - twsinst 400
- agent uninstalling
 - twsinst 422, 424
- agent upgrade
 - scanning system prerequisites 48, 236
- agent upgrade return code
 - twsinst 400, 400
- AgentCertificateDownloader script
 - deploying certificates to agents 482
 - deploying certificates to dynamic agents 482
 - deploying certificates to fault-tolerant agents 482
 - downloading certificates to agents 482
- agents
 - authorization to install 115
 - direct upgrade 255
 - parallel upgrade 366
 - upgrading 255, 289, 366
- agents uninstalling
 - twsinst 422
- AI Data Advisor 182
- AIDA 182

- Amazon EKS
 - moving to 389
- Amazon Web Services 176, 177, 178, 179, 183
- API Key authentication in upgrade 369
- application job plug-ins
 - option to add runtime for Java runtime to run job types with advanced options 226, 292, 297
 - option to add the Java runtime to run job types with advanced options using twsinst 225
- applications
 - workload environment integrated with 22
- AWS 176, 177, 178, 179, 182, 183
- AWS CloudFormation 178
- AWSJIM1001W error
 - installing or upgrading on a Windows 307
- AWSRES003E error message 403

B

- back-level MDM
 - certificate management 371
- backup dir too small
 - installing or upgrading on a Windows 308
- backup domain manager
 - configuring 213
 - installation parameters 442
- backup
 - dynamic domain manager 245
 - configuring 215
 - custom certificates 154
 - environment 12
 - uninstalling 419
 - backup master domain manager
 - configuring 210
 - direct upgrade 246
 - environment 12
 - in 9.4 environment 364
 - in back-level environment 364
 - install 359
 - uninstalling 416
 - backup master installation 442
- batchman
 - checking if active 412
- bdm installation 442
- before installing
 - multiple agent instances 300
- BKDDM
 - custom certificates 154
- bkm installation 442
- bottom-up upgrade 228

C

- capability
 - dynamic agent 12
 - dynamic domain manager 12
 - extended agent 13
 - fault-tolerant agent 13, 13
- Centralized agent update
 - agent is in running status 310, 310
- Centralized update
 - multiple agent instances 300
- certificate conversion 371
 - before upgrading from 9.4 258, 313
 - certificate conversion
 - before upgrading from 9.5 258, 313

- certificates 375
 - downloading for different user 487
- cloud
 - deployment 176
- CloudFormation 178
- cluster
 - deployment 178
- cluster environment 179
- command-line installation 45
 - using properties files 96
- commands
 - twsinst to add the Java runtime to run job types with advanced options 225
- commands and scripts
 - ps, used before manual uninstallation 412
 - shut, used before manual uninstallation 412
 - stop
 - used before manual uninstallation 412
 - unlink
 - used before manual uninstallation 412
 - wdlssp, used before manual uninstallation 412
 - wdrmvsp, used before manual uninstallation 412
- configureDb script
 - configuration 430
 - database configuration 430
 - database population 430
 - schema creation 430
- configuring
 - backup domain manager 213
 - backup
 - dynamic domain manager 215
 - backup master domain manager 210
 - domain manager 212
 - dynamic agent 216
 - dynamic domain manager 214
 - fault-tolerant agent 140
 - master domain manager 208
 - z-centric agent 224
- connection from TDWC
 - engine connection fails 406
- containers 374, 375
- Containers
 - Deploying with Docker 171
- converting certificates from JKS to PEM 371
- converting default certificates 371
- creating
 - stacks 178
- credentials 179
- custom certificates 200
 - upgrading 239, 258
- customizing
 - chart 183

D

- database certificates 95
- database migration procedure
 - Dynamic Workload Console 237
- database properties file 96
- database schema
 - upgrade 268, 338
- database update 94

- database upgrade error
 - error when upgrading a DB2 database 229, 406
- Db2
 - SSL mode 95
 - using certificates 95
- Db2 license
 - applying 46
 - upgrading 46
 - using with current installation 46
- DB2
 - prerequisite
 - for
 - master domain manager 45
- DDM
 - custom certificates 154
- ddm installation 442
- default certificates 375
 - converting before upgrading 258, 313
 - default certificates
 - upgrade procedure 258, 313
 - extracting before upgrading 258, 313
 - upgrading 378
 - upgrading with different product versions 371
- deleting files
 - too slow after manual uninstall 414
- deploy 43
- Deploying
 - AWS 176
 - with Docker compose 171
- depot directory
 - populating with certificates 371
- depot folder 371
- direct updating 245
- direct upgrade 228
 - version 110.
 - x
 - .
 - x
 - to version 10.2.5 238
 - version 9.5.0.
 - x
 - to version 10.2.5 238
- directories created outside of TWA_home
 - when installing
 - HCL Workload Automation 41
- distributed workload
 - environment 16
 - environment with dynamic scheduling capabilities 17, 24
 - environment with static and dynamic scheduling capabilities 20
- distributed-driven
 - workload environment for z/OS 23
- docker 374, 375
- Docker compose
 - prerequisites 171
- Docker containers
 - master domain manager installation 173
- docker image
 - master domain manager installation
 - method 43
- dockerfile 176
- domain

- amount of network traffic 27
- dependencies between jobs 27
- firewalls 28
- internetwork dependencies 29
- level of fault-tolerance required 28
- localized processing 26
- number of geographic locations 27
- number of workstations, applications, and jobs 27
- planning 26, 26
- system performance and other criteria 27
- time zones 27, 27
- topology
 - multiple 31
 - single 28
- types of applications 27
- Windows network 27
- domain manager
 - configuring 212
- domain managers
 - direct upgrade 255
 - parallel upgrade 366
 - upgrading 255, 289, 366
- download center 182
- downloading
 - Agent 182
 - OCLI 182
 - templates 177
- DWC 182
 - monitoring query problems 228
- DWC certificates 95
- DWC data
 - exporting to file 237
- DWC JDBC drivers
 - updating 94
- DWC settings
 - exporting to file 237
 - importing 336
- dwcinst script
 - Dynamic Workload Console 456
- dynamic agent
 - capability 12
 - configuring 216
 - environment 12
 - gateway 135
 - gateway parameters 119, 465
 - installing
 - authorization requirements 115
 - dockerfile 176
 - on 115
- Dynamic Agent 182
- dynamic and static scheduling capabilities
 - environment with 20
- dynamic domain manager
 - 154, 245
 - configuring 214
 - custom certificates 154
 - environment 12
 - SSL configuration 203
 - uninstalling 419, 421
- dynamic domain manager
 - installation
 - scanning system prerequisites 48, 236
- dynamic scheduling
 - enabling 225
- dynamic scheduling capabilities
 - environment with 17, 24
- Dynamic Workload Console 182
 - accessibility ix
 - create database 319

- database creation 319
- dwcinst script 456
- engine connection
 - fails after downgrading 406
 - uninstalling 418
- Dynamic Workload Console
 - data
 - exporting to file 237
 - migrating to a new database 237
 - migrating to a new node 237
- Dynamic Workload Console
 - Db2 certificates 95
- Dynamic Workload Console
 - JDBC drivers
 - updating 94
- Dynamic Workload Console
 - PostgreSQL certificates 95
- Dynamic Workload Console
 - settings
 - exporting to file 237
 - importing 336
 - migrating to a new database 237
 - migrating to a new node 237

E

- enabling
 - dynamic scheduling 225
- enabling TLS 1.2
 - upgrading from v 9.4 312
- encryption upgrade error
 - error when encrypting useropts file 407
- end-to-end scheduling 34
- end-to-end workload environment
 - planning 22
- engine connection from TDWC
 - fails after returning from FP1 to 9.5 GA 406
- environment
 - backup
 - dynamic domain manager 12
 - backup master domain manager 12
 - description 10
 - distributed workload environment 16
 - distributed workload environment with dynamic scheduling capabilities 17, 20, 24
 - distributed-driven workload environment for z/OS 23
 - domain 26
 - dynamic agent 12
 - dynamic domain manager 12
 - end-to-end workload environment 22
 - extended agent 13
 - localized processing 26
 - master domain manager 11
 - workload environment integrated with external systems 22
- environment static
 - fault-tolerant agent 13
 - standard agent 13
- environment variables
 - setting 206
- error when upgrading a DB2 database
 - database upgrade error 229, 406
- exporting
 - repository data 237
 - repository settings 237
- extended agent
 - capability 13

- environment 13
- EXTENDED_ROW_SZ DB2 option 229, 406
- external systems
 - workload environment integrated with 22

F

- fault-tolerant agent
 - configuring 140
 - static capability 13
- feature
 - adding new 225
- File Proxy installation 478
- File Proxy start 480
- File Proxy stop 481
- File Proxy uninstallation 481
- fileproxyinst script 478
 - File Proxy installation 478
- fileproxystart script 480
 - File Proxy start 480
- fileproxystop script 481
 - File Proxy stop 481
- files
 - /etc/password 294
 - FINAL 208
 - Symphony 33
 - TWSRegistry.dat 412
- FINAL
 - adding 208
- final job stream
 - adding 208
- FIPS compliance
 - PKCS#12 keystores 379
- FIPS prerequisites 379
- FIPS requirements 379
- from version 9.5.0.
- x
- or 10.
- x
- .
- x
- to version
- 10.2.5
- parallel upgrade 257

G

- gateway
 - installation parameters 119, 465
 - introduction 10
- generating SQL files
 - database setup 90
- getting
 - credentials 179
- getting started 177
- Github
 - templates 177

H

- HCL Workload Automation
 - directories created outside of TWA_home
 - at installation time 41
 - installation path 35
- HCL Workload Automation agent
 - 113
- HCL Workload Automation scanning
 - system prerequisites for
 - HCL Workload Automation
 - 48, 236
- HCL Workload Automation service for TWS_user
 - deleting 410

- helm chart
 - deployment 178
- Helm Chart
 - template 177
- HWA 176, 183
- HWA on
 - AWS 176, 177, 178, 179, 182, 182, 183, 184

I

- importing certificates into DWC keystore 110
- install
 - backup master domain manager
 - 359
 - Java runtime 119, 159, 225, 290, 297, 442, 465
- installation 119, 119, 119, 465, 465, 465
 - agent 113
 - checking prerequisites IBM i 158
 - directories created outside of TWA_home
 - when installing
 - HCL Workload Automation
 - 41
 - gateway 135
 - log files 397
 - scanning system prerequisites for
 - HCL Workload Automation
 - 48, 236
 - troubleshooting 397
- Installation
 - images 232
 - on your workstation 232
- installation agent
 - return code 400
- installation and uninstallation log files
- twinsinst 399
- installation images
 - downloading 45
- installation method
 - twinsinst 113
- installation methods
 - docker image for master domain manager 43
- installation user
 - keeping track of 116
 - retrieving 116
- installing
 - backup dir too small 308
 - error AWSJIM1001W 307
- Installing fix packs or upgrading
 - multiple agent instances 300
- installing from the CLI 45
- installing master domain manager
 - Docker containers 173
- integrating
 - AIDA 182
- interface
 - command line client 14
 - dynamic workload broker
 - command line
 - 14
 - Dynamic Workload Console
 - 14
 - master domain manager command line 14
- internetwork dependencies
 - domain 29

J

- Java runtime
 - corrupted registry 225
 - installation 119, 159, 225, 290, 297, 442, 465
 - recover 225

- registry file
 - recovery 225
- JDBC drivers
 - customizing 94
 - replacing 94
 - settings 94
 - updating 94
- JDBC drivers download 94
- JKS certificates
 - converting 371
 - upgrading 371
- job count not sent to MHS server 52
- jobman and JOBMAN
 - checking if active 412
- JWT
 - downloading for different user 487
- JWT problems
 - upgrade MDM with custom certificates 234
 - upgrading to V10.1 Fix Pack 1 or later 234
- jwt.crt creation
 - DDM and BKDDM custom certificates 154

L

- language packs
 - installing 108, 125, 162, 293, 434, 447, 461, 471
- Liberty
 - configuration changes 199
 - data_dir 199
 - separated configuration 199
 - server.xml 199
- Liberty upgrade 239, 260, 315
- license management
 - MHS 52
- licensing
 - MHS 52
 - procedure 52
 - troubleshooting 52
- Linux user accounts 49
- local option descriptions
 - SSL encryption cipher 201
 - SSL version 201
- localized processing
 - domain 26
- log files 397

M

- mailman
 - checking if active 412
- manual uninstall
 - agents 410
 - master domain manager
 - 410
- master domain manager
 - configuring 208
 - direct upgrade 251
 - environment 11
 - installation parameters 442
 - prerequisite 45
 - SSL configuration 203
 - switching to upgraded backup master 249
 - uninstall manually 410
 - uninstalling 417
- master domain manager
 - Db2 certificates
 - 95
 - master domain manager
 - installation
 - scanning system prerequisites 48, 236
- master domain manager
 - installation
 - scanning system prerequisites 48, 236
- JDBC drivers
 - updating 94

- master domain manager
- PostgreSQL certificates
 - 95
- master domain manager upgrade
 - API Key authentication 369
 - scanning system prerequisites 48, 236
- master installation method
 - serverinst for master domain manager 43
- master upgrade
 - API Key authentication 369
- MDM certificates 95
- mdm installation 442
- MDM JDBC drivers
 - updating 94
- MDM upgrade
 - API Key authentication 369
- MHS
 - permanent license 52
 - term license 52
- MHS licensing 52
- MHS server
 - connection procedure 52
 - connection verification 52
 - troubleshooting 52
- migrating from on-prem to cloud 389
- migrating from on-prem to Kubernetes 389
- mixed-version environment
 - upgrading with default certificates 371
- modify
 - option to add the Java runtime to run job types with advanced options using twsinst 225
- moving databases 237
- moving from Db2 to PostgreSQL 237
- MSSQL DB
 - creation error 408
- multiple agent instances
 - Centralized update 300
 - creating
 - before installing 300
 - Installing fix packs or updating 300

N

- netman
 - checking if active 412
- Netman for TWS_user, deleting service 410
- network 10
 - backup
 - dynamic domain manager
 - 12
 - backup master domain manager 12
 - dynamic agent 12
 - dynamic domain manager
 - 12
 - extended agent 13
 - master domain manager 11
 - network static
 - agent fault-tolerant 13
 - standard agent 13
- new backup master domain manager
 - parallel upgrade 348
- new Db2 license 46
- new MDM
 - SSL connection to existing DWC 110
- new MDM, existing DWC
 - SSL connection 110
- No Monitor Operator Messages
 - Centralized agent update 310
- no root 115
- no-root agent installation 115

- no-root installation 100, 150, 267, 337

O

- OCLI 182
- OpenShift
 - moving to 389
- OpenSSL
 - SSL encryption cipher, local option 201
- OpenSSL 3.0.x libraries
 - support on RHEL 9 or later 403
- operator message
 - Centralized agent update 310, 310
 - update does not complete 310, 310
- optional password encryption 427
- Oracle E-Business Suite applications
 - workload environment integrated with 22
- oracle partitioning feature
 - EDWA improvement 91
 - event-driven workload automation improvement 91
 - performance improvement 91
- Oracle
 - prerequisite
 - for
 - master domain manager
 - 45
- Orchestration CLI 182

P

- packages
 - download 182
- parallel upgrade 228
 - backup master domain manager
 - installation
 - 360
 - from 9.4 403
 - from 9.5 403
 - from version 9.5.0.
 - x
 - or 10.
 - x
 - .
 - x
 - to version
 - 10.2.5
 - 257
 - new backup master domain manager 348, 360
 - version 9.4.0.
 - x
 - to version
 - 10.2.5
 - 311
 - parallel upgrade from 9.4
 - TLS 1.2 312
 - parameter twsinst
 - modify 226
 - parameter twsinst modify
 - acceptlicense 226
 - addjruntime 226
 - inst_dir 226
 - password 227
 - recovInstReg 227
 - uname 227
 - parameter twsinst update
 - addjruntime 292, 297
 - inst_dir 292
 - lang 293
 - password 293
 - reset_perm 293
 - skip_usercheck 294
 - tdwbhostname 299

- tdwbport 299
- uname 294, 299
- update 294, 299
- wait 294, 299
- work_dir 299
- password encryption 58
- PEM certificates
 - agent security 482
 - converting to 371
 - upgrading to 371
- Peoplesoft applications
 - workload environment integrated with 22
- planning
 - distributed workload environment 16
 - distributed workload environment with dynamic scheduling capabilities 17, 24
 - distributed workload environment with static and dynamic scheduling capabilities 20
 - distributed-driven workload environment for z/OS 23
 - domain 26, 26
 - end-to-end workload environment 22, 22
 - environment 16, 17, 20, 24
 - localized processing in your domain 26
 - workload environment integrated with external systems 22, 22
- post installation
 - configuring a backup domain manager 213
 - configuring backup
 - dynamic domain manager
 - 215
 - configuring backup master domain manager 210
 - configuring domain manager 212
 - configuring dynamic agent 216
 - configuring
 - dynamic domain manager
 - 214
 - configuring fault-tolerant agent 140
 - configuring master domain manager 208
 - configuring
 - z-centric agent
 - 224
 - PostgreSQL
 - SSL mode 95
 - using certificates 95
 - PostgreSQL database
 - collate command 409
 - collate feature 409
 - incorrect behaviour 409
 - troubleshooting 409
 - prerequisite
 - master domain manager
 - 45, 171
 - prerequisite Docker deployment
 - master domain manager
 - 45, 171
 - prerequisite scan
 - error AWSJIM1001W 307
 - prerequisites
 - IBM i 158
 - ps, command used before manual
 - uninstallation 412

R

- registry entries, deleting manually
 - UNIX 412
 - Windows 410
- registry file
 - recreating 370

- upgrading with corrupt files 370
- remote command-line client
 - configuration 221
 - installation 113
- removing the product
 - dynamic domain manager 421
 - twinsinst 422, 424
- REST service error 403
- restore agent
 - return code 400
- return code
 - twinsinst 400, 400, 400, 400, 400
- RHEL 9 or later
 - OpenSSL 3.0.x libraries 403
 - SHA-1 signatures 403

S

- SAP R/3 applications
 - workload environment integrated with 22
- scale down 43
- scale up 43
- scan
 - system prerequisites for HCL Workload Automation 48, 236
- scan prerequisite
 - error AWSJIM1001W 307
- scanning
 - system prerequisites for HCL Workload Automation 48, 236
- schema creation
 - configureDb script 430
- secure script 427
 - optional password encryption 427
- security 200
 - encrypting passwords 58
 - password decryption 58
 - security
 - decrypting passwords 58
- security certificates 371
- serverinst
 - agent installation method 43
 - master domain manager installation method 43
- serverinst script
 - backup domain manager installation 442
 - backup dynamic domain manager installation 442
 - dynamic domain manager for a Z controller installation 442
 - dynamic domain manager installation 442
 - master domain manager installation 442, 442
- services (Windows)
 - deleting 410
- shut, command, used before manual
- uninstallation 412
- Single Sign-On
 - configuring 195
- software prerequisites
 - verifying 233
- SQL files review
 - database setup 90

- ssl 119, 465
- SSL
 - OpenSSL, SSL encryption cipher, local option 201
- SSL configuration
 - dynamic domain manager 203
 - enforcing after upgrading 378
 - master domain manager 203
- SSL connection
 - new
 - master domain manager, existing
 - Dynamic Workload Console 110
 - new MDM, existing DWC 110
- SSL connection to existing DWC
 - new MDM 110
- SSL encryption cipher, local option 201
- SSL mode
 - update does not complete 310
- SSL version
 - enabling using local option 201
- SSL version, local option 201
- SSO
 - configuring 195
- stack 178
- Stacks 178
- stageman
 - checking if active 412
- standard agent
 - capability static 13
 - environment static 13
- static and dynamic scheduling capabilities
 - environment with 20
- static capability
 - fault-tolerant agent 13
 - standard agent 13
- static network
 - domain manager 13
- step
 - configuring a backup domain manager 213
 - configuring backup dynamic domain manager 215
 - configuring backup master domain manager 210
 - configuring domain manager 212
 - configuring dynamic agent 216
 - configuring dynamic domain manager 214
 - configuring fault-tolerant agent 140
 - configuring master domain manager 208
 - configuring z-centric agent 224
- stop, command
 - used before manual uninstallation 412
- subscribing to
 - HWA on AWS 177
- subscribing to Workload Automation on Amazon Web Services 177
- switching databases 237
- switching from Db2 to PostgreSQL 237
- Symphony file 33
- syntax
 - twinsinst to add the Java runtime to run job types with advanced options 225
- system prerequisites

- scan for
 - HCL Workload Automation 48, 236
- systems external
 - workload environment integrated with 22

T

- template 178
- test connection to engine from TDWC fails after version reversal 406
- time zone
 - overview 28
- tls 200
- tls 1.2 200
- TLS 1.2
 - upgrading from v 9.4 312
- tls 1.3 200
- Token Service
 - for TWS_user, deleting service 410
- top-down upgrade 228
- troubleshooting
 - installation 397
- twins_env file customization
 - upgrading 229
- twinsinst 113, 115
 - installation and uninstallation log files 166, 300, 399, 425
 - installation method 113
 - return code 400, 400, 400, 400, 400
 - syntax to add the Java runtime to run job types with advanced options 225
 - uninstalling 422, 424
 - UNIX usage 226, 291
 - Windows usage 226, 291
- TWSRegistry.dat, file 412
- TWSUser
 - deleting from registry
 - UNIX 412
 - Windows 410

U

- uninstall
 - manually
 - agents 410
 - master domain manager 410
- uninstall script 481
 - file proxy uninstallation 481
- uninstallation
 - as no-root user 416
 - manual
 - file deletion too slow 414
 - the main components 416
 - troubleshooting 397
 - user requirements 416
- uninstallation agent
 - return code 400
- uninstalling
 - backup dynamic domain manager 419
 - backup master domain manager 416
 - dynamic domain manager 419, 421
 - Dynamic Workload Console 418
 - master domain manager 417
- Uninstalling
 - HWA
 - chart 184
- uninstalling agent
 - twinsinst 422, 424
- UNIX
 - uninstalling manually 412
- UNIX user accounts 49

- unlink, command
 - used before manual uninstallation 412
- update 245, 374, 375
- updating containers 374, 375
- updating Db2 license 46
- upgrade
 - backup master domain manager 359
 - bottom-up 228
 - considerations 228
 - database schema 268, 338
 - implications 228
 - mixed-level environments 228
 - scanning system prerequisites 48, 236
 - top-down 228
 - troubleshooting 397
 - verifying software prerequisites 233
- upgrade agent
 - return code 400, 400
- upgrade MDM with custom certificates
 - JWT problems 234
- upgrade problem on RHEL 9 or later
 - default certificates 403
- upgrade questions 378
- upgrading
 - backup dir too small 308
 - chart 183
 - default certificates 378
 - enabling API Key authentication 369
 - error AWSJIM1001W 307
 - fault-tolerant agent 370
 - with corrupt registry files 370
- upgrading a dynamic domain manager
 - certificate conversion 258, 313
- upgrading from 9.4
 - certificate conversion 258, 313
- upgrading from 9.5
 - certificate conversion 258, 313
- upgrading from v 9.4 312
 - ensuring communication 312
- upgrading Liberty 239, 260, 315
- upgrading to V10.1 Fix Pack 1 or later
 - JWT problems 234
- upgrading
 - WebSphere Application Server Liberty 260, 315
 - upgrading
 - WebSphere Application Server Liberty Base 239
 - upgrading with default certificates 371
 - user is not db admin 95
 - user other than root user
 - agent installation 115
 - useropts upgrade error
 - encryption upgrade error 407
- users
 - TWS_user
 - deleting from registry on UNIX 412
 - deleting from registry on Windows 410

V

- variables
 - symlink
 - TWA/TWS/bin/at 34
 - TWA/TWS/bin/batch 34
 - TWA/TWS/bin/datecalc 34
 - TWA/TWS/bin/jobstdl 34
 - TWA/TWS/bin/maestro 35
 - TWA/TWS/bin/mdemon 35
 - TWA/TWS/bin/morestdl 35
 - TWA/TWS/bin/muser 35

- TWA/TWS/bin/parms 35
- verifying software prerequisites
 - upgrade 233
- version 10.
 - x
 - .
 - x
 - to version 10.2.5
 - direct upgrade 238
 - version 9.4.0
 - x
 - to 10.2.5
 - parallel upgrade 311
 - version 9.5.0.
 - x
 - to version 10.2.5
 - direct upgrade 238

W

- wdlssp, command used before manual uninstallation 412
- wdrmvsp, command used before manual uninstallation 412
- WebSphere Application Server Liberty
 - upgrading 260, 315
- WebSphere Application Server Liberty Base
 - upgrading 239
- WebSphere Application Server prerequisite
 - for master domain manager 45
- WebSphere SDK Java Technology Edition prerequisite
 - for master domain manager 45
- Windows
 - file deletion to slow after manual uninstallation 414
 - uninstalling manually 410
- Windows systems
 - backup dir too small when installing or upgrading 308
 - error AWSJIM1001W installing or upgrading 307
- workload automation 176
- Workload Automation
 - home installation path 35
- workload on the cloud 389
- Workload Scheduler agents IBM i uninstalling
 - twinst 424
- workstation class
 - definition 28
- writer
 - checking if active 412

Z

- z-centric agent
 - configuring 224
- z/OS applications
 - workload environment integrated with 22